



**UNIVERSIDAD TÉCNICA DEL
NORTE**



INSTITUTO DE POSTGRADO

Maestría en Ingeniería de Software

**INTEGRACIÓN DE ORACLE Y WORDPRESS PARA LA
ACTUALIZACIÓN DE CONTENIDOS EN EL PORTAL WEB DE
LA UNIVERSIDAD TÉCNICA DEL NORTE**

**Trabajo de grado previo a la obtención del Título de Magíster en
Ingeniería de Software**

AUTORA

Doris Andrea Arcos Ponce

TUTOR

Ing. Pedro Granda MSc.

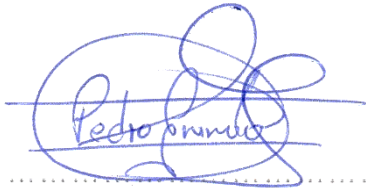
Ibarra - Ecuador

2017

Aprobación del tutor

En mi calidad de tutor del trabajo de grado: INTEGRACIÓN DE ORACLE Y WORDPRESS PARA LA ACTUALIZACIÓN DE CONTENIDOS EN EL PORTAL WEB DE LA UNIVERSIDAD TÉCNICA DEL NORTE, presentado por Doris Andrea Arcos Ponce, para optar por el grado de Magíster en Ingeniería de Software, doy fe de que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a presentación y evaluación por parte del jurado examinador que se designe.

Ibarra, noviembre de 2017.



Ing, Pedro Granda MSc.

TUTOR DEL TRABAJO DE GRADO

Carta de aceptación Tribunal

Ibarra, 14 de noviembre de 2017

Magíster
Jorge Caraguay
DIRECTOR POSGRADO UTN

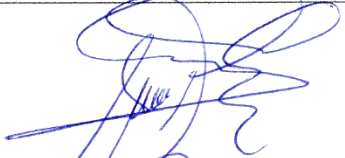
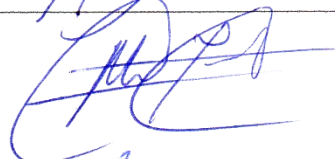
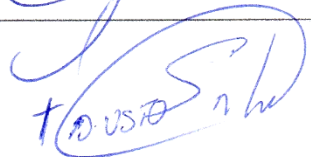
De nuestras consideraciones:

Nos permitimos informar a usted que revisado el Trabajo de Grado de la maestrante Doris Andrea Arcos Ponce, del Programa de Maestría en Ingeniería de Software, con el tema **“INTEGRACIÓN DE ORACLE Y WORDPRESS PARA LA ACTUALIZACIÓN DE CONTENIDOS EN EL PORTAL WEB DE LA UNIVERSIDAD TÉCNICA DEL NORTE”** tenemos a bien certificar que han sido acogidas y satisfechas todas las observaciones realizadas en la defensa privada.

En tal virtud, facultamos empastar el mencionado trabajo y que su tutor solicite fecha para defensa pública.

Agradecemos su atención.

Atentamente,

	Apellidos y Nombres	Firma
Miembro Tribunal 1	MSc. Marco Pusdá	
Miembro Tribunal 2	MSc. Pablo Landeta	
Miembro Tribunal 3	MSc. Fausto Salazar	

Carta de aceptación auspiciante



UNIVERSIDAD TECNICA DEL NORTE

Universidad Acreditada resolución 002-CONEA-2010-129-DC
Resolución No. 001-073-CEAACES-2013-13

DIRECCION DE DESARROLLO TECNOLOGICO E INFORMATICO

CARTA DE ACEPTACION

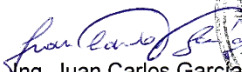
DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INFORMÁTICO

19 de septiembre de 2017.

Magíster
Jorge Caraguay Procel
DIRECTOR POSGRADO UTN

Me permito informar a usted que la ingeniera: ARCOS PONCE DORIS ANDREA con número de cédula 0401255518, estudiante del Programa de Maestría en: Ingeniería de Software, ha concluido de manera eficiente y satisfactoria el proyecto de investigación **“INTEGRACIÓN DE ORACLE Y WORDPRESS PARA LA ACTUALIZACIÓN DE CONTENIDOS EN PORTAL WEB DE LA UNIVERSIDAD TÉCNICA DEL NORTE”** aplicado en la Dirección de Desarrollo Tecnológico e Informático de la UTN. Además debo informar que el sistema se encuentra implementado y funcionando correctamente.

Atentamente,


Ing. Juan Carlos García
DIRECTOR DDTI



Visión Institucional

La Universidad Técnica del Norte en el año 2020, será un referente en ciencia, Tecnología e innovación en el país, con estándares de excelencia internacionales.

Autorización de uso y publicación a favor de la Universidad Técnica del Norte

1. Identificación de la Obra

La Universidad Técnica del Norte dentro del proyecto de Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad. Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

Datos de contacto			
Cédula de identidad:	0401255518		
Apellidos y nombres:	Arcos Ponce Doris Andrea		
Dirección:	Cdla. La Laguna 1 Mz. D C. # 5 Tulcán – Carchi – Ecuador		
Correo electrónico:	daap1313@gmail.com		
Teléfono fijo:	062224803	Teléfono móvil:	0999311007

Datos de la Obra	
Título:	INTEGRACIÓN DE ORACLE Y WORDPRESS PARA LA ACTUALIZACIÓN DE CONTENIDOS EN EL PORTAL WEB DE LA UNIVERSIDAD TÉCNICA DEL NORTE
Autora:	Arcos Ponce Doris Andrea
Fecha AAAA/MM/DD:	2017-11-14
Programa:	Posgrado
Título por el que opta:	Magíster en Ingeniería de Software
Asesor/Director:	Ing. Pedro Granda MSc.

2. Autorización de uso a favor de la Universidad

Yo, Doris Andrea Arcos Ponce, con cédula de identidad N° 0401255518, en calidad de autora y titular de los derechos patrimoniales del trabajo de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo

a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

3. Constancia

La autora manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 14 días del mes de noviembre de 2017.

Autora



Doris Andrea Arcos Ponce

CC. 0401255518

Aceptación



Ing. Betty Chávez

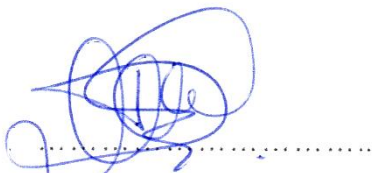
Biblioteca UTN

**Cesión de derechos de autor del trabajo de grado a favor de la Universidad
Técnica del Norte**

Yo, Doris Andrea Arcos Ponce, con cédula de identidad N°. 0401255518, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autora de la obra o trabajo de grado denominado “INTEGRACIÓN DE ORACLE Y WORDPRESS PARA LA ACTUALIZACIÓN DE CONTENIDOS EN EL PORTAL WEB DE LA UNIVERSIDAD TÉCNICA DEL NORTE”, que ha sido desarrollado para optar por el título de Magister en Ingeniería de Software, en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autora me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

Ibarra, a los 14 días del mes de noviembre de 2017.

LA AUTORA

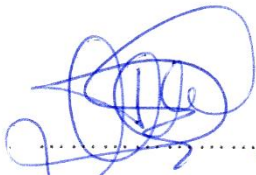


Doris Andrea Arcos Ponce

CC. 0401255518

Autoría

Yo, Doris Andrea Arcos Ponce, portadora de la cédula de ciudadanía N° 0401255518 declaro que la presente investigación denominada “INTEGRACIÓN DE ORACLE Y WORDPRESS PARA LA ACTUALIZACIÓN DE CONTENIDOS EN EL PORTAL WEB DE LA UNIVERSIDAD TÉCNICA DEL NORTE”, es de mi autoría y responsabilidad, y se han respetado las diferentes fuentes de información, realizando las citas correspondientes.



.....
Doris Andrea Arcos Ponce

CC. 0401255518

Dedicatoria

*A mi familia por su incondicional amor,
ejemplo de vida y voz de aliento,
por inspirar en mí el
deseo constante de superación.*

Agradecimiento

*Mi sincera gratitud a todas aquellas personas
quienes con sus valiosos aportes y buenos augurios
posibilitaron la consecución de este trabajo.*

Doris Andrea Arcos Ponce

Resumen

La integración de aplicaciones surge como una necesidad para organizaciones que cuentan con ecosistemas de software heterogéneo caracterizados por la coexistencia de aplicaciones disímiles entre sí, en donde la información organizacional se almacena en diferentes fuentes de datos; la integración en el nivel más sencillo busca el intercambio de información.

La propuesta presentada en este documento corresponde al software que vincula el Sistema Informático Integrado Universitario (SIIU) –plataforma Oracle– con el blog de la Carrera de Ingeniería de Sistemas Computacionales (CISIC) –WordPress– del portal web de la Universidad Técnica del Norte (UTN). El software fue desarrollado siguiendo el marco de trabajo Scrum, los sprints definidos permitieron implementar las funcionalidades requeridas; cada sprint fue ejecutado después de su respectiva planificación, al completarse se verificó el cumplimiento de tareas y resultados logrados.

En términos generales la vinculación del SIIU y blog de la CISIC se efectuó mediante dos módulos, el primero enfocado a la exposición de la información de interés del SIIU mediante API's Rest y el segundo encargado de la actualización de contenidos en el blog haciendo uso de la API Rest de WordPress.

Los servicios web son una alternativa para integrar aplicaciones con el fin de compartir información entre ellas. La propuesta desarrollada permite actualizar contenidos seleccionados de manera automática en el blog de la CISIC aprovechando la información disponible en el SIIU, constituyendo una herramienta que apoya las tareas de gestión de contenidos en el blog.

Palabras clave: Integración de aplicaciones, WordPress, Sistema Informático Integrado Universitario, Portal web, servicio web basado en Rest, WordPress Rest API.

Abstract

Application integration emerges as a necessity for organizations that have heterogeneous software ecosystems characterized by the coexistence of applications with each other, where organizational information is stored in different data sources; integration at the simplest level seeks the exchange of information.

The proposal presented in this document corresponds to the software that links the Sistema Informático Integrado Universitario (SIIU) - Oracle platform - to the blog of Carrera de Ingeniería de Sistemas Computacionales (CISIC) - WordPress - of the web portal of the Universidad Técnica del Norte (UTN). The software was developed following the framework of the Scrum work, the sprints allowed to implement the required functionalities; each sprint was executed after their respective planning, completing the accomplishment of the tasks and results achieved.

In general terms, the linkage of the SIIU and the CISIC's blog was done through two modules, the first focused on the presentation of information of interest of the SIIU through the API Rest and the second in charge of the update of contents in the blog with use WordPress Rest API.

Web services are an alternative for integral applications in order to share information between them. The developed proposal allows to update the results of the automatic form in the blog of the CISIC, taking advantage of the information available in the SIIU, constituting a tool that supports the tasks of the management of contents in the blog.

Keywords: Application Integration, WordPress, Sistema Informático Integrado Universitario, Web portal, Restful web service, WordPress Rest API.

Índice general

Aprobación del tutor	ii
Carta de aceptación Tribunal	iii
Carta de aceptación auspicante	iv
Autorización de uso y publicación a favor de la Universidad Técnica del Norte	v
Cesión de derechos de autor del trabajo de grado a favor de la Universidad Técnica del Norte	vii
Dedicatoria.....	ix
Agradecimiento	ix
Resumen	x
Abstract.....	xi
Índice general	xii
Índice de figuras	xv
Índice de tablas	xvii
CAPÍTULO I	1
1. EL PROBLEMA	1
1.1. Contextualización del problema	1
1.2. Planteamiento del problema	2
1.3. Formulación del problema.....	2
1.4. Justificación.....	3
1.5. Objetivos.....	4
1.5.1. Objetivo general.....	4
1.5.2. Objetivos específicos	4
CAPÍTULO II	5
2. MARCO REFERENCIAL	5
2.1. Marco Teórico	5
2.1.1. Integración de aplicaciones.....	5
2.1.1.1. Niveles de integración.....	8
2.1.1.2. Integración de datos	11
2.1.1.3. Infraestructura/Topologías de integración de aplicaciones	14
2.1.2. Servicios web	19
2.1.2.1. Servicios web basados en SOAP.....	21
2.1.2.1.1. Características	23
2.1.2.1.2. Arquitectura.....	25
2.1.2.2. Servicios REST	27
2.1.2.2.1. Operaciones empleadas en REST	28
2.1.3. WordPress.....	34
2.1.3.1. API REST de WordPress	38
2.1.4. Node.js	40

2.1.4.1. LoopBack	41
CAPÍTULO III	44
3. MARCO METODOLÓGICO	44
3.1. Lugar de desarrollo de la investigación	44
3.1.1. Unidad Ejecutora	44
3.1.2. Ubicación	46
3.1.3. Beneficiarios	46
3.2. Modalidad de investigación.....	46
3.3. Nivel o tipo de investigación	46
3.4. Población y muestra	46
3.5. Operacionalización de variables	47
3.6. Métodos	49
3.7. Técnicas e instrumentos.....	49
3.8. Recolección de la información	50
3.9. Análisis e interpretación de resultados	50
3.9.1. Resultados de la encuesta.....	51
3.9.2. Resultados de las entrevistas.....	58
3.9.2.1. Entrevista aplicada al Web máster institucional	58
3.9.2.2. Entrevista realizada al Director del DDTI.....	59
3.9.2.3. Entrevista a los administradores de los módulos del SIIU.....	60
3.9.3. Revisión de contenidos de los blogs	61
3.9.4. Conclusiones y recomendaciones	68
CAPÍTULO IV	70
4. PROPUESTA	70
4.1. Antecedentes de la propuesta	70
4.2. Metodología de desarrollo de software	72
4.3. Integración del SIIU y el blog de la CISIC (Solución propuesta)	74
4.3.1. Objetivos.....	75
4.3.2. Arquitectura de integración	75
4.3.3. Esquema del software	77
4.4. Pila de producto.....	79
4.5. Detalle de la ejecución de sprints	80
4.5.1. Planificación del sprint	80
4.5.2. Ejecución del sprint	82
4.5.2.1. Análisis.....	82
4.5.2.2. Codificación	83
4.5.2.3. Pruebas	91
4.6. Sprints backlogs.....	92
4.6.1. Sprint 0.....	93
4.6.2. Sprint 1.....	96

4.6.3. Sprint 2.....	99
4.6.4. Sprint 3.....	102
4.6.5. Sprint 4.....	105
4.7. Plan de pruebas.....	107
4.8. Revisión de sprints	115
4.9. Beneficios de la propuesta.....	120
4.10. Resultados del proyecto	121
CAPÍTULO V	128
5. CONCLUSIONES Y RECOMENDACIONES	128
5.1. Conclusiones.....	128
5.2. Recomendaciones	129
BIBLIOGRAFÍA	131
Anexos.....	135
Anexo 1 – Revisión de contenidos informativos - Blogs de carreras de la UTN.....	135
Anexo 2 – Cuestionarios de encuestas y entrevistas	137
A.- Entrevista dirigida al web máster	137
B.- Entrevista dirigida al Director de Desarrollo Tecnológico e Informático.....	137
C.- Entrevista dirigida a los administradores de módulos del SIIU	138
D.- Encuesta aplicada a docentes y estudiantes.....	140
E.- Encuesta aplicada a Stakeholders	141

Índice de figuras

Figura 1. Integración a nivel de datos	9
Figura 2. Integración a nivel de interfaz de usuario	9
Figura 3. Integración a nivel de método	10
Figura 4. Integración a nivel de aplicación.....	11
Figura 5. Replicación de base de datos.....	12
Figura 6. Federación de datos.....	13
Figura 7. Topología de integración punto a punto.....	15
Figura 8. Topología de integración bróker	16
Figura 9. Topología de bus	17
Figura 10. Tendencias en la evolución de la integración de aplicaciones empresariales	18
Figura 11. Servicio web.....	20
Figura 12. Servicio web basado en SOAP.....	22
Figura 13: Roles en un servicio web	25
Figura 14: Pila de protocolos de un servicio web.....	26
Figura 15. Interacción cliente y servidor en RESTful	29
Figura 16: Diagrama de la base de datos de WordPress.....	35
Figura 17: Módulos de LoopBack y sus dependencias	42
Figura 18. Organigrama del departamento de Desarrollo Tecnológico e Informático...	45
Figura 19. Encuesta – Pregunta 1	51
Figura 20. Encuesta – Pregunta 2	52
Figura 21. Encuesta – Pregunta 3	53
Figura 22. Encuesta – Pregunta 4	53
Figura 23. Encuesta – Pregunta 5	54
Figura 24. Encuesta – Pregunta 6	55
Figura 25. Encuesta – Pregunta 7	56
Figura 26. Encuesta – Pregunta 8	57
Figura 27: Revisión de blogs de carrera	62
Figura 28: Tipos de contenidos en los blogs de carrera	62
Figura 29: Porcentaje de ocurrencia de los contenidos	63
Figura 30. Menús del blog de la CISIC	70
Figura 31. Arquitectura de integración.....	76
Figura 32. Esquema del software	78
Figura 33. Código de API Rest Misión	84
Figura 34. Función que actualiza el contenido Misión y visión.....	88
Figura 35. Contenido Misión y visión del blog.....	89
Figura 36. Datos devueltos por las API's Rest Misión y visión.....	90
Figura 37. Misión y visión de la CISIC almacenadas en el SIIU	91
Figura 38. Burndown de Sprint 0	95

Figura 39. Burndown de Sprint 1	98
Figura 40. Burndown de Sprint 2	101
Figura 41. Burndown de Sprint 3	104
Figura 42. Burndown de Sprint 4	107
Figura 43. Tabulación – Pregunta 1	124
Figura 44. Tabulación – Pregunta 2	125

Índice de tablas

Tabla 1. Operaciones CRUD HTTP	29
Tabla 2. Principales códigos de estado HTTP – Categoría 1XX.....	30
Tabla 3. Principales códigos de estado HTTP – Categoría 2XX.....	31
Tabla 4. Principales códigos de estado HTTP – Categoría 3XX.....	31
Tabla 5. Principales códigos de estado HTTP – Categoría 4XX.....	32
Tabla 6. Principales códigos de estado HTTP – Categoría 5XX.....	32
Tabla 7. Estructura de la base de datos de WordPress	36
Tabla 8. Referencia para desarrolladores de los endpoints de la API REST de WP.....	40
Tabla 9. Principales módulos de LoopBack	43
Tabla 10. Población	47
Tabla 11. Operacionalización de variables.....	48
Tabla 12. Tabulación – Encuesta Pregunta 1	51
Tabla 13. Tabulación – Encuesta Pregunta 2	51
Tabla 14. Tabulación – Encuesta Pregunta 3	52
Tabla 15. Tabulación – Encuesta Pregunta 4	53
Tabla 16. Tabulación – Encuesta Pregunta 5	54
Tabla 17. Tabulación – Encuesta Pregunta 6	55
Tabla 18. Tabulación – Encuesta Pregunta 7	56
Tabla 19. Tabulación – Encuesta Pregunta 8	57
Tabla 20. Blogs de carreras por facultad	58
Tabla 21. Revisión de contenidos del blog de la CISIC	64
Tabla 22. Product Backlog	79
Tabla 23. Funcionalidades definidas para cada Sprint	80
Tabla 24. Información a vincular entre las aplicaciones	82
Tabla 25. API's Rest implementadas	85
Tabla 26. Funciones implementadas en el módulo de actualización de información	87
Tabla 27. Backlog de Sprint 0	94
Tabla 28. Backlog de Sprint 1	97
Tabla 29. Backlog de Sprint 2	100
Tabla 30. Backlog de Sprint 3	103
Tabla 31. Backlog de Sprint 4	106
Tabla 32. Plan de pruebas – Sprint 1	108
Tabla 33. Plan de pruebas – Sprint 2	110
Tabla 34. Plan de pruebas – Sprint 3	112
Tabla 35. Plan de pruebas – Sprint 4	113
Tabla 36. Revisión de resultados alcanzados - Sprint 1	116
Tabla 37. Revisión de resultados alcanzados – Sprint 2.....	117
Tabla 38. Revisión de resultados alcanzados - Sprint 3	118

Tabla 39. Revisión de resultados alcanzados - Sprint 4	119
Tabla 40. Características de la propuesta de integración implementada.....	121
Tabla 41. Tabulación encuesta stakeholders	123
Tabla 42. Tabla cruzada.....	126
Tabla 43. Pruebas de chi-cuadrado.....	126

CAPÍTULO I

1. EL PROBLEMA

1.1. Contextualización del problema

El Mandato Constituyente N° 14 de 2008 dispuso al Consejo Nacional de Evaluación y Acreditación (CONEA) la elaboración de un “informe técnico sobre el nivel de desempeño institucional de los establecimientos de educación superior, a fin de garantizar su calidad, propiciando su depuración y mejoramiento” (Derecho Ecuador, 2008). A partir de entonces las instituciones de educación superior (IES) han sido sometidas a varios procesos de evaluación externa, como el efectuado en el año 2013 en el cual el Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior (CEAACES) en el área de Tecnologías de la Información y comunicación (TIC) evaluó la innovación tecnológica, es decir lo correspondiente al “... número de procesos académicos que se encuentran automatizados, ... considerando los siguientes: sistema de matrículas, registro académico, registro y consulta de notas, currículo académico, blogs para docentes, aulas virtuales, materiales de estudio” (CEAACES, 2013, pág. 41), evidenciando la importancia de las tecnologías de la información y comunicación (TIC) como herramientas al servicio de la comunidad universitaria.

Dos de las aplicaciones informáticas con las que cuenta la Universidad Técnica del Norte (UTN) son el portal web y el Sistema Informático Integrado Universitario (SIIU). En el portal web se publican noticias, información sobre la oferta académica, además incluye accesos a sistemas informáticos relacionados con el quehacer administrativo y académico de la institución. El SIIU por su parte, cuenta con módulos que gestionan los procesos académicos, financieros, de planificación institucional, biblioteca, órganos colegiados, recaudación, contabilidad, entre otros.

El portal web y el SIIU son aplicaciones independientes soportadas por servidores, lenguajes de programación y bases de datos de diferentes proveedores; debido a ello surge la imposibilidad de compartir información de manera transparente entre ambas.

1.2. Planteamiento del problema

En el portal web universitario se encuentra publicada la oferta académica de la UTN tanto a nivel de grado como de posgrado. Cada carrera cuenta con un blog informativo cuyos contenidos generales como información, gestión académica, investigación, vinculación y contactos para la mayoría de casos se encuentran estandarizados.

La información de cada carrera como ficha, misión y visión, objetivos educacionales, resultados de aprendizaje, malla curricular, docentes investigadores/producción científica, trabajos de titulación, convenios y listados de graduados, entre otra, está publicada en el portal web y también consta en el SIIU. Cuando la información mencionada es modificada en el SIIU se debería realizar una tarea similar en el gestor de contenidos del portal web para que la información contenida en ambos sea la misma, situación que no siempre ocurre enseguida, ocasionando que los contenidos publicados en el portal no estén actualizados oportunamente.

La obtención de información desde una aplicación para emplearla en otra, es una tarea complicada pues requeriría la intervención de técnicos informáticos especializados que conozcan las bases de datos y aplicaciones, realicen un proceso de exportación/importación de datos, además de la revisión de los esquemas de bases de datos para vincular la información de interés. Por lo expuesto anteriormente es conveniente plantear una solución encaminada a compartir datos entre las aplicaciones mencionadas, en busca mejorar los procesos de actualización de contenidos en el portal web de la UTN.

1.3. Formulación del problema

La gestión de contenidos en el portal web es una tarea que no logra cumplirse a tiempo ocasionando que la información publicada en él no se encuentre actualizada oportunamente.

1.4. Justificación

El portal web y el SIIU son aplicaciones informáticas que cubren diferentes necesidades institucionales, existe información similar que es gestionada por cada aplicación de forma independiente en su propio gestor de base de datos. Considerando el escenario descrito, la investigación propuesta es importante porque aborda una temática de interés: la integración de aplicaciones; área trascendente no sólo para la institución en mención sino para otras que cuentan con una realidad similar en cuanto al tipo de aplicaciones informáticas.

Esta investigación es de utilidad porque busca solucionar un problema práctico al integrar aplicaciones de características diferentes, generando como resultado la actualización de contenidos en el portal web; constituyéndose en una herramienta de apoyo para los administradores del mismo.

La propuesta a desarrollar beneficiará directamente a la UTN ya que permitirá optimizar tiempos en cuanto a la actualización de contenidos en el portal web gracias a que se aprovechará la información disponible en el SIIU; dicha tarea comprenderá inicialmente la identificación de aquellos ítems de información publicados en el portal web que se encuentran almacenados en la base de datos del SIIU con el respectivo detalle del módulo y/o tabla que los gestiona; para posteriormente establecer la factibilidad de ser compartidos entre las aplicaciones; y por último implementar la solución que permita compartir la información seleccionada entre el SIIU y el portal web.

La investigación es de impacto ya que permitirá establecer un proceso general de integración con la finalidad de compartir información, mismo que podría aprovecharse con otras aplicaciones entre las que se requiera una tarea similar; y es factible porque existe el interés de parte de las autoridades, además de contar con los recursos tanto tecnológicos, humanos y económicos para llevarse a cabo.

1.5. Objetivos

1.5.1. Objetivo general

Integrar Oracle y Wordpress para la actualización de contenidos del portal web de la Universidad Técnica del Norte.

1.5.2. Objetivos específicos

- Describir los fundamentos teóricos de la integración de aplicaciones, niveles y tecnologías de integración.
- Diagnosticar la situación de la gestión de contenidos en el portal web de la UTN para proponer una alternativa de integración entre éste y el SIIU.
- Implementar una aplicación que permita la actualización de contenidos informativos del blog de la Carrera de Ingeniería en Sistemas Computacionales, a partir de información proveniente del SIIU.

CAPÍTULO II

2. MARCO REFERENCIAL

2.1. Marco Teórico

En una organización es común la existencia de aplicaciones que brindan funcionalidades determinadas y fueron incorporadas en distintos tiempos; tal situación implica que varias de esas aplicaciones se hayan desarrollado por distintos equipos de personas y lenguajes de programación, con diferentes gestores de bases de datos, e inclusive para diferentes plataformas de hardware y software. Toda esta mixtura tecnológica se presenta debido a que cada aplicación fue implantada en un determinado momento para cubrir un requerimiento en la empresa, resultando en aplicaciones aisladas que gestionan información relacionada pero alojada en diferentes almacenes de datos.

Rosa Sequeira, Frantz, Yevseyeva, Emmerichd y Basto-Fernandes (2015) definen al escenario descrito como “ecosistema de software heterogéneo”, conformado por diferentes tipos de aplicaciones que incluyen sistemas heredados, software de terceros o desarrollos internos. En dicho ecosistema la integración no fue considerada cuando las aplicaciones fueron diseñadas; por tanto, como lo señalan Moturi, Githehu y Kahonge (2013) éstas no pueden comunicarse entre sí, dando lugar a problemas como duplicación de información, integridad de datos y hasta incoherencia en informes generados. Modificar cada aplicación no es una opción, lo conveniente es encontrar mecanismos que permitan utilizar las aplicaciones y bases de datos existentes.

2.1.1. Integración de aplicaciones

La integración de aplicaciones fue planteada como una de las soluciones a la problemática anteriormente explicada, para Oracle también es conocida como integración de aplicaciones empresariales (EAI – Enterprise Application Integration) y la define así:

Es el proceso de vincular aplicaciones para simplificar y automatizar procesos empresariales. En el nivel más básico implica el intercambio de datos y funciones de una aplicación de software a otra, incluidas aplicaciones empaquetadas,

aplicaciones personalizadas y muchos tipos de sistemas de información heredados. (2014, pág. 9)

Según Linthicum (2003) “la integración de aplicaciones es un enfoque estratégico para vincular muchos sistemas de información en conjunto, tanto en el nivel de servicio como de información, apoyando su capacidad de intercambiar información y aprovechar procesos en tiempo real”.

La integración de aplicaciones surge de la realidad tecnológica de organizaciones que cuentan con aplicaciones informáticas heterogéneas, las definiciones citadas indican que la integración implica vincular aplicaciones independientemente de plataformas y tecnologías; de manera que sea posible intercambiar información y compartir funcionalidades para apoyar procesos empresariales aprovechando las aplicaciones existentes.

Mediante una solución de integración de aplicaciones es posible sincronizar datos entre varias aplicaciones o incorporar nuevas funcionalidades sobre ellas, de manera que no sean afectadas por la solución de integración (Rosa Sequeira et al., 2015). Para Baskaran (2014) la principal ventaja de la integración es conectar aplicaciones sin que haya la necesidad de implementar cambios drásticos en ellas, para lograrlo es esencial tener una clara idea del proceso del negocio así como de la información de interés que se quiere compartir.

Fu (2016) afirma que la información es un activo valioso para las organizaciones, que puede incidir positivamente en la productividad del negocio e inclusive generar beneficios de carácter económico a los clientes; de allí que éstas centren sus esfuerzos en sacar el máximo provecho a su información que generalmente está disponible en fuentes diversas. Para Fuseau (2015) la necesidad de integración surge debido a que una organización puede pasar por etapas de expansión, contracción, adquisición o fusión; que requieren enormes esfuerzos en el área de automatización de procesos para responder a los cambios organizacionales. Tanto para empresas pequeñas como grandes “se ha

convertido en una prioridad de misión crítica conectar aplicaciones dispares y aprovechar su colaboración en toda la empresa con el fin de mejorar la eficiencia general del negocio, mejorar la escalabilidad y reducir los costos de TI” (MuleSoft, 2014).

La integración de aplicaciones apoya los siguientes objetivos empresariales (California Department of Technology, 2014):

- **Mejoras en productividad y flexibilidad.**- La integración de aplicaciones permite brindar funciones y servicios empresariales de manera ágil y a un menor costo, en comparación al uso de aplicaciones individuales o integradas de forma limitada.
- **Mejoras en la calidad de datos y procesos.**- Las herramientas para integración de aplicaciones permiten identificar y eliminar redundancia en datos, funciones y/o procesos; también ayudan a compartir funcionalidades entre aplicaciones, sistemas o áreas funcionales.
- **Facilitar fusiones organizacionales y reemplazos del sistema.**- La integración simplifica tanto el proceso de fusión o sustitución de aplicaciones internas, como la integración con aplicaciones externas a la organización.

La integración de aplicaciones no es un área de interés reciente, ha sido una necesidad desde que las organizaciones cuentan con redes de computadoras y más de una aplicación informática para la gestión de sus operaciones; lo que ha evolucionado a través del tiempo es la visión de la integración de aplicaciones como herramienta de apoyo a las estrategias de negocio. La tendencia es el paso de la integración orientada a datos hacia la orientada a servicios, la primera es una opción económica que cubre múltiples dominios y que generalmente no involucra realizar cambios en las aplicaciones; la segunda en cambio es compleja pues implica vincular servicios, requiere la modificación de las aplicaciones existentes e inclusive la creación de otras, sin embargo es obvio que le da mayor valor a la empresa (Linthicum, 2003).

Para SPAN Systems Corporation (2014) la integración de aplicaciones debe hacer frente a varios retos explicados a continuación:

- **Interoperabilidad.**- Surge debido a problemas de compatibilidad entre plataformas, lenguajes, datos y formatos de datos; dificultando la conexión a través de una interfaz estándar.
- **Integración de datos.**- Una de las principales funcionalidades de una solución de integración de aplicaciones es la gestión del flujo de información entre aplicaciones asegurando su coherencia.
- **Robustez, estabilidad y escalabilidad.**- Son características presentes en las aplicaciones a nivel individual, por lo tanto deben mantenerse en las aplicaciones de integración.

El Departamento de Tecnología de California (2014) además de los retos a nivel técnico menciona otros en el ámbito organizacional ocasionados por la divergencia de objetivos de las funciones y procesos empresariales, cuyo funcionamiento era óptimo a nivel individual pero no al integrarse.

2.1.1.1. Niveles de integración

Cada organización debe enfrentar sus propios retos en referencia a la integración de aplicaciones debido a los objetivos empresariales perseguidos y a las características de las aplicaciones a integrar. Según Baskaran (2014) los niveles de integración son:

- **Nivel de datos.**- La integración es llevada a cabo para compartir información entre almacenes de datos como bases de datos, gestores de archivos, textos e imágenes. En este escenario pueden verse involucradas varias bases de datos con cientos de tablas. Desde una base de datos se toma información, se procesa y se actualiza en otra, comúnmente se utilizan herramientas ETL (Extract, Transform, Load – Extraer, Transformar, Cargar), tal como se presenta en la Figura 1. Para este tipo de integración no se necesita realizar modificaciones significativas en las aplicaciones existentes, además de los bajos costos de implementación ya que emplea middleware orientados a bases de datos (JDBC/ODBC).

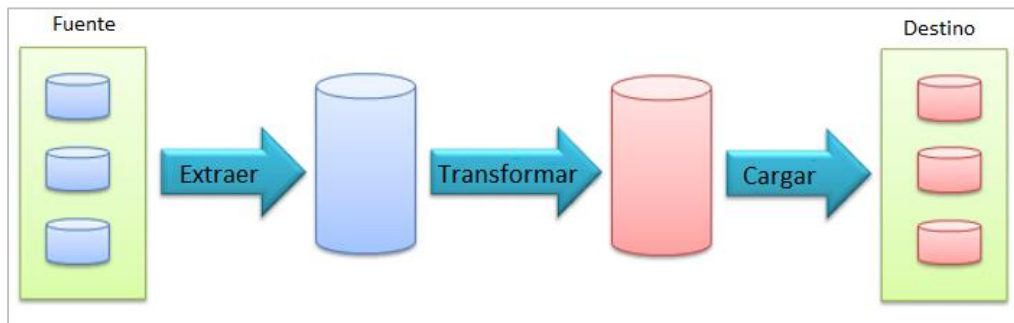


Figura 1. Integración a nivel de datos

Fuente: (Mudrakola, 2017)

- Nivel de interfaz de usuario.**- La integración de aplicaciones a este nivel implica la definición de interfaces de usuario para obtener datos, localizarlos, extraerlos, leerlos y procesarlos o almacenarlos en otra interfaz de usuario (Figura 2). La integración se da cuando dos o más aplicaciones comparten información a través de interfaces de usuario. Esta implementación demanda que se conozcan las aplicaciones a ser integradas junto a su esquema de datos y lógica de negocio, debido a que no siempre coincide los datos presentados en pantalla con la forma en que se almacenan en la base de datos. Con la integración a nivel de interfaz de usuario se persigue rastrear la información desde la interfaz de usuario hacia la base de datos mediante la lógica de la aplicación.

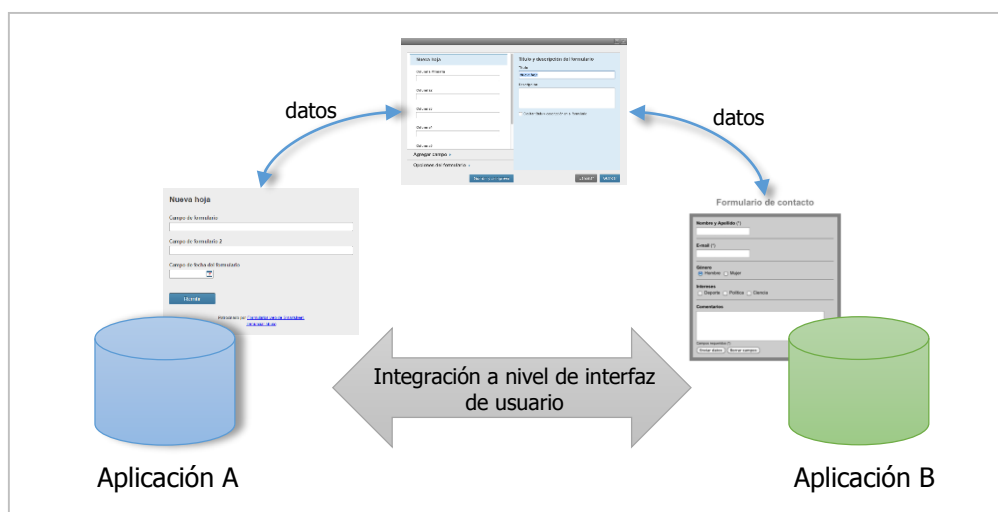


Figura 2. Integración a nivel de interfaz de usuario

Fuente: Elaboración propia.

- Nivel de método.**- “Un método o un proceso es cualquier regla de negocio o una parte de lógica que existe dentro de una empresa que tiene un efecto sobre cómo se procesa la información” (Baskaran, 2014, pág. 33). La integración a nivel de método (Figura 3) proporciona una infraestructura que permite a las aplicaciones acceder a un servidor que contiene métodos compartidos, es decir métodos de la lógica del negocio común para las aplicaciones. En este nivel de integración, tecnologías como Java, RMI, CORBA, DCOM son ampliamente utilizadas; aunque los servicios web es uno de los enfoques más seguidos por los proveedores de software. Para llevar a cabo la integración a este nivel, las aplicaciones participantes deben ser modificadas para que puedan acceder a los métodos compartidos, lo que resulta en mayores costos de implementación; sin embargo compartir la lógica de negocio entre varias aplicaciones es de enorme valía para una organización. Este nivel de integración es similar al nivel de aplicación, pero a un nivel de granularidad menor.

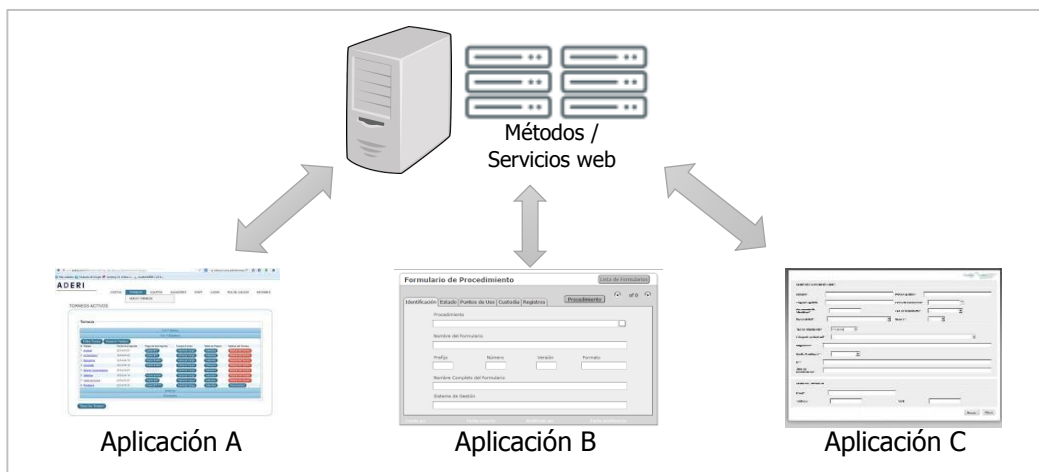


Figura 3. Integración a nivel de método

Fuente: Elaboración propia.

- Nivel de aplicación.**- En este nivel de integración se aprovechan las APIs (Application Programming Interface) provistas por las aplicaciones. Una API define los procedimientos para acceder a los servicios de una aplicación, mismos que pueden ser: procesos de negocio, de datos y objetos. Los procesos de negocio brindan acceso a la lógica del negocio de la aplicación. Los servicios de datos

posibilitan el acceso al almacén de datos con el objetivo de extraer información, las actualizaciones usualmente se llevan a cabo a través de servicios de negocio. Los objetos encapsulan servicios de negocio y de datos. “Es el tipo de integración más típico seguido por los proveedores de software para desarrollar aplicaciones empresariales” (Baskaran, 2014, pág. 34).

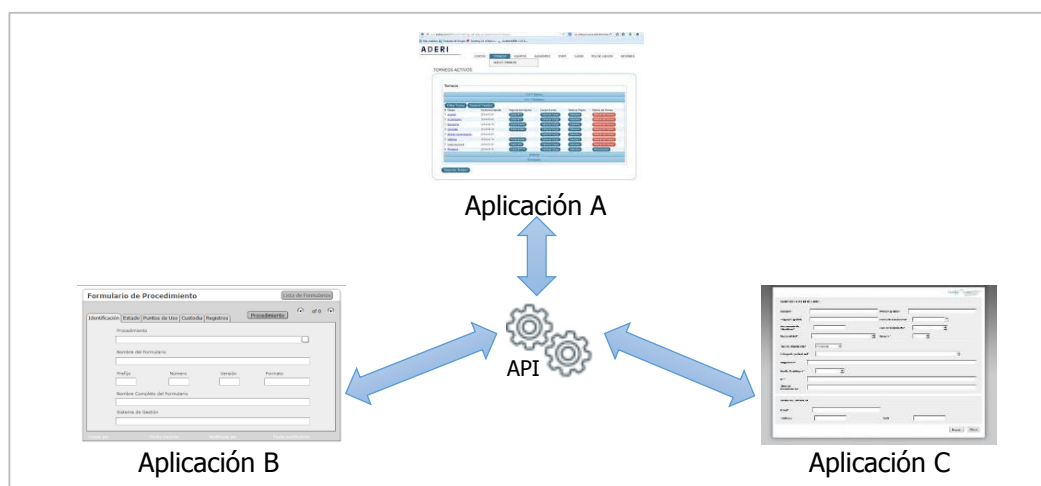


Figura 4. Integración a nivel de aplicación

Fuente: Elaboración propia.

La selección del método adecuado de integración depende de los objetivos que persigue la empresa y de su situación tecnológica. Cada aplicación debe ser analizada con el fin de identificar sus interfaces potenciales, si la aplicación no tiene APIs entonces podría considerarse la integración a nivel de almacenes de datos; en caso contrario la integración a nivel de aplicación podría ser la opción a elegir (Lin, 2005).

2.1.1.2. Integración de datos

La integración de información organizacional es una tarea comúnmente complicada, pues ésta reside en distintos gestores de bases de datos, con características tecnológicas particulares y que gestionan esquemas específicos.

Consolidar la información en un sistema de base de datos es complejo, debido a que una aplicación que se ejecuta contra una determinada base de datos no lo puede hacer

transparentemente contra otra. Pese a que existen semejanzas entre bases de datos relacionales; se debe recalcar también que las diferencias son significativas en cuanto a capacidades propias, estructura de diccionarios de datos, tipos de datos, dialecto SQL y semántica de transacciones SQL (Oracle, 2009).

La integración de datos o integración orientada a la información se efectúa entre almacenes de datos, el punto de integración son las bases de datos o APIs que producen información. Según Linthicum (2003) al respecto dos de los principales enfoques son:

- **Replicación de datos:** Consiste en el intercambio de datos entre dos o más bases de datos que pueden pertenecer a diferentes proveedores (Figura 5). La infraestructura de replicación permite solventar las diferencias entre modelos y esquemas de las bases de datos. En el mercado existen soluciones middleware que posibilitan este proceso, éstas incluyen herramientas de transformación para vincular datos desde la fuente origen con datos en la fuente destino. Entre sus ventajas sobresalen el bajo costo y la simplicidad.

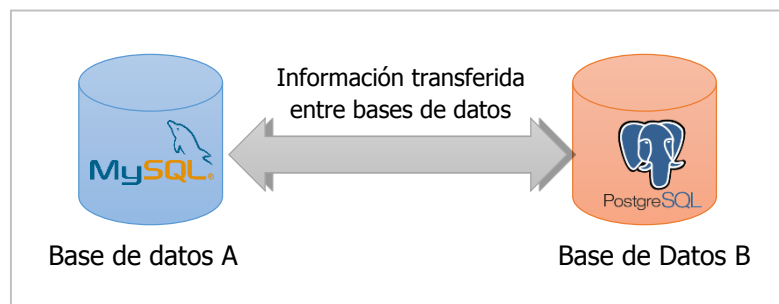


Figura 5. Replicación de base de datos

Fuente: Adaptado de (Linthicum, 2003, pág. 32)

La Figura 5 representa a dos bases de datos de diferentes proveedores entre las que se está efectuando un proceso de replicación de datos, esta tarea podría requerirse por ejemplo cuando se realiza la actualización de datos en un determinado sistema informático y se necesita que dicho cambio se propague en la base de datos de otra aplicación.

- Federación de datos:** Es la integración de varias bases de datos y modelos en una base de datos virtual (Figura 6). Las herramientas para federación de datos incluyen una capa middleware entre las bases de datos físicas y las aplicaciones que acceden a los datos, esta capa a través de interfaces se conecta a las bases de datos físicas asignándoles un modelo de base de datos virtual. La principal ventaja es que permite conectar varias bases de datos en un modelo que permite el intercambio de información a través de una interfaz definida. En esta solución se deben realizar cambios a nivel de aplicación para que admita software de base de datos federada, debido a que se utilizan varias interfaces para acceder a la base de datos virtual.

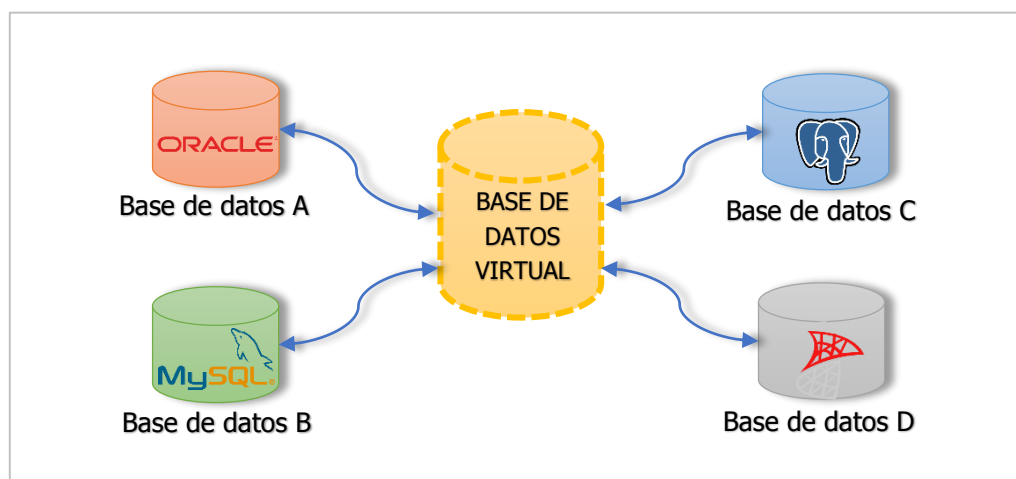


Figura 6. Federación de datos

Fuente: Adaptado de (Linthicum, 2003, pág. 33)

En la Figura 6 varias bases de datos (Oracle, MySQL, PostgreSQL y SQL Server) a través de la federación se muestran como si se tratara de única base de datos global. Este escenario podría presentarse por ejemplo para el caso de dos empresas que cuentan con sus propias bases de datos y se fusionan, para formar parte de una misma organización.

Solutions Review (2017) indica que la integración de datos comprende procesos técnicos y empresariales que permiten combinar datos provenientes de diversas fuentes para convertirlos en información valiosa para la organización. El software de integración

soporta tareas de procesamiento analítico contra un datawarehouse, permitiendo a los usuarios finales alinear, combinar y obtener datos. ETL y otros procesos como la replicación son empleados en la gestión de datos. Las empresas más sobresalientes en el área de integración de datos son: Actian, Adeptia, Altova, Attivio, Attunity, Denodo, Dell Boomi, HVR Software, IBM, Informatica, Information Builders, Jitterbit, Liaison Technologies, Microsoft, MuleSoft, Oracle, Pentaho, SAP, SAS, Software AG, Stone Bond Technologies, Striim, Syncsort, Talend. Solutions Review incluye diez preguntas enfocadas a la selección de una herramienta para integrar datos:

1. ¿Por qué es necesaria una herramienta de integración de datos?
2. ¿Qué tipo de datos se van a analizar?
3. ¿Cuáles son las fuentes de datos y dónde están ubicadas?
4. ¿Se requiere cloud, on-premise o ambos?
5. ¿Están consideradas las características de calidad de datos y gestión de datos maestros?
6. ¿La herramienta es compatible con la infraestructura existente?
7. ¿Es la flexibilidad un elemento crucial de su solución de integración?
8. ¿Se integra con Hadoop, Spark y otros repositorios NoSQL?
9. ¿La solución viene con algún servicio de soporte?
10. ¿Cuál es la principal área de enfoque que mejor hace su herramienta? (págs. 4-5-6)

2.1.1.3. Infraestructura/Topologías de integración de aplicaciones

La integración de aplicaciones ha ido evolucionando conforme lo han hecho las aplicaciones en una organización. Inicialmente consistió en la vinculación entre dos o más aplicaciones; sin embargo, cuando creció la cantidad y complejidad de las mismas los requerimientos a ser considerados han dado lugar a diferentes propuestas en cuanto a topologías o infraestructuras, siendo las principales:

- **Topología punto a punto.**- En esta topología se implementa un componente de conexión por cada par de aplicaciones que se desee vincular; mismo que se

encarga de la transformación de datos, integración y cualquier otro servicio requerido. La topología punto a punto se recomienda cuando la cantidad de aplicaciones a integrar es pequeña resultando en una solución ligera que cubre las necesidades de una organización; sin embargo a medida que crece el número de aplicaciones a integrar también lo hace de manera exponencial la cantidad de componentes de conexión (SPAN Systems Corporation, 2014).

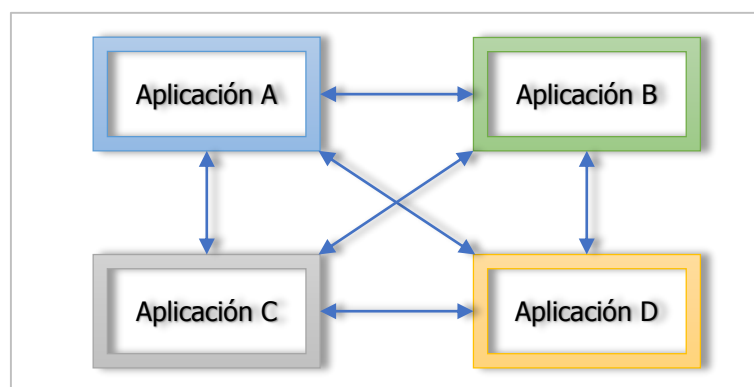


Figura 7. Topología de integración punto a punto

Fuente: Adaptado de (SPAN Systems Corporation, 2014, pág. 4)

La Figura 7 representa a las aplicaciones de una organización integradas mediante una topología punto a punto, ellas se comunican directamente entre sí. Du, Peng, y Zhou (2008) manifiestan que para organizaciones que tienen una gran cantidad de aplicaciones el modelo punto a punto termina siendo una conexión espagueti; sin embargo, permite la comunicación síncrona y persistente entre aplicaciones.

- **Topología de bróker (Hub/Spoke).**- Utiliza un punto central de control (hub) encargado de formatear mensajes, enrutarlos y proporcionar otras tareas requeridas por las aplicaciones. En esta topología también están presentes herramientas de monitoreo, auditoría, y otras que asisten en la tarea de asignación y enrutamiento entre aplicaciones. Con esta topología existe un bajo acoplamiento entre aplicaciones, la configuración de integración se realiza en un repositorio central; sin embargo el hub es un punto de fallo y con altas cargas de trabajo puede convertirse en un cuello de botella (SPAN Systems Corporation, 2014).

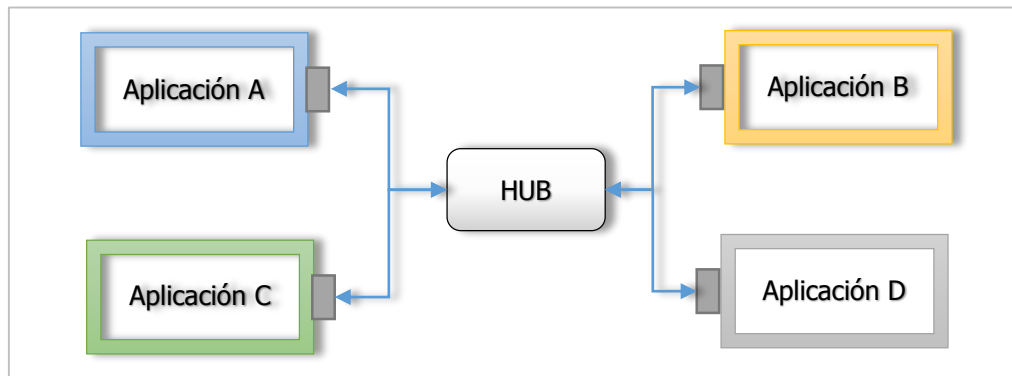


Figura 8. Topología de integración bróker

Fuente: Adaptado de (SPAN Systems Corporation, 2014, pág. 4)

En la Figura 8 el hub recibe los mensajes enviados por las aplicaciones, se encarga de formatearlos y retransmitirlos, es un único punto central de control que con excesiva carga podría convertirse en un punto de fallo.

- **Topología de bus.**- Mediante la topología de bus se busca eliminar los problemas de sobrecarga, cuellos de botella y punto único de fallos presentados por la topología de bróker, para ello las tareas se dividen y asignan a varios componentes. Una de las principales ventajas de la topología de bus es que se logran “soluciones de integración ligeras y personalizadas con fiabilidad garantizada, totalmente abstraídas de la capa de aplicación, siguen un patrón consistente” (SPAN Systems Corporation, 2014, pág. 5). El diseño y configuración de soluciones de integración que siguen la topología de bus no requiere que las aplicaciones a integrar sean modificadas.

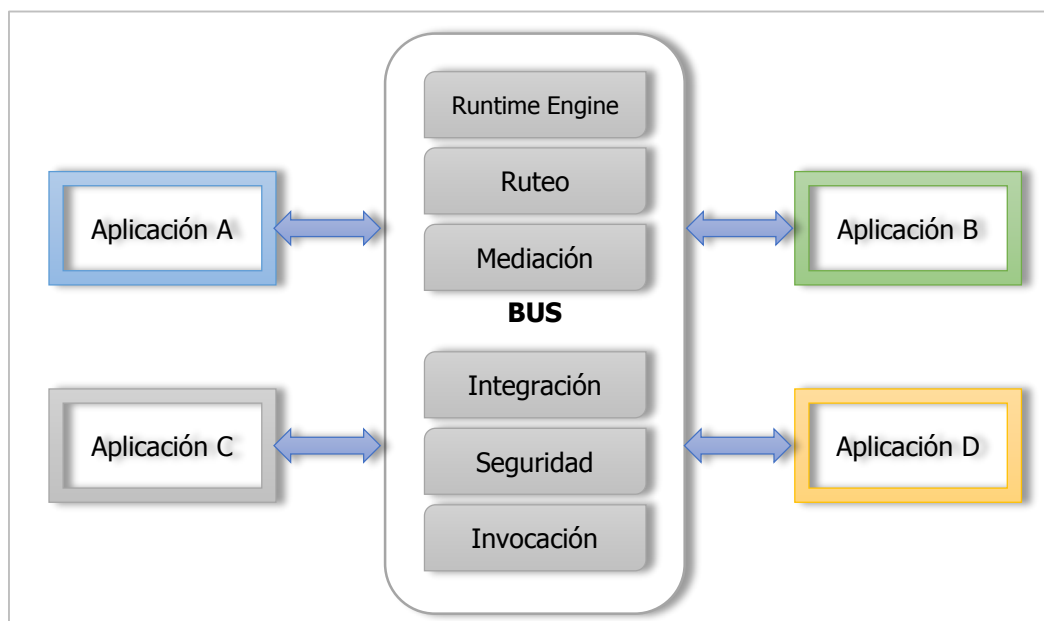


Figura 9. Topología de bus

Fuente: Adaptado de (SPAN Systems Corporation, 2014, pág. 6)

La Figura 9 representa la integración de aplicaciones a través de una topología de bus; como lo explican Du, Peng y Zhou (2008) las aplicaciones origen colocan mensajes en el bus accesibles al resto de aplicaciones, las aplicaciones destino se suscriben según requieran a los mensajes emitidos en el bus.

La versión madura de esta topología es el Bus de Servicios Empresariales (ESB – Enterprise Services Bus) que constituye el elemento central de la integración a gran escala, encargado del transporte fiable de datos y mensajes entre productores y consumidores. Sus componentes principales son (California Department of Technology, 2014, págs. 13-14):

- **Servicio de registro y repositorio.**- Almacena la información sobre los servicios disponibles, quedando almacenada para su recuperación e identificación.
- **Enrutamiento de mensajes.**- Enruta un mensaje determinado del remitente a uno o múltiples consumidores, en función del mensaje y las reglas del negocio externas.

- **Servicio de invocación.**- Posibilita la invocación de un tipo particular de servicio, en base a los datos de invocación e información sobre la interfaz del servicio.
- **Motor de procesos de negocio/flujo de trabajo.**- Proporciona la ejecución de definiciones de proceso del negocio declaradas y validadas.
- **Agregaciones y almacenamiento temporal.**- Respalda aspectos técnicos del funcionamiento del ESB, tales como entrega garantizada o transformaciones de carga útil de datos específicos.

Según SPAN Systems Corporation (2014) se identifican tres tendencias en la evolución de la integración de aplicaciones empresariales (Figura 10): integración tradicional, integración de aplicaciones empresariales e integración de servicios.

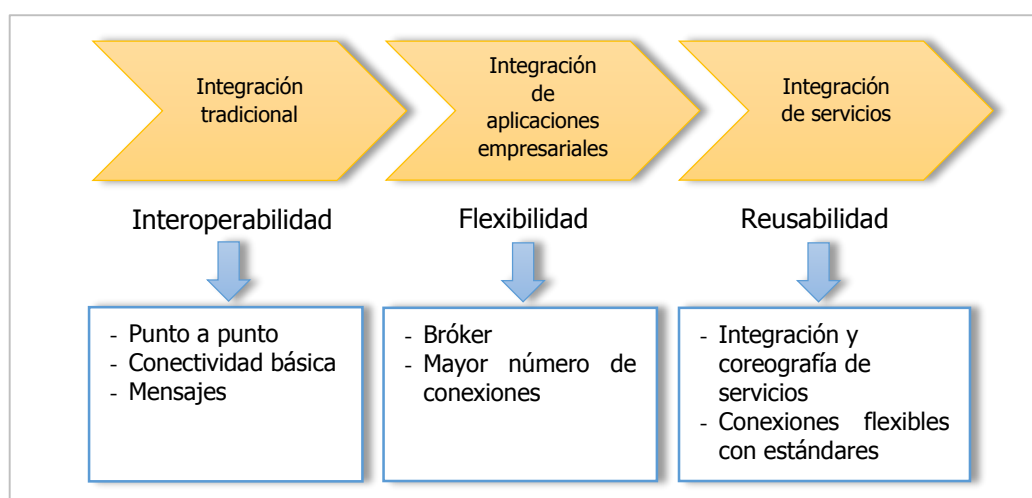


Figura 10. Tendencias en la evolución de la integración de aplicaciones empresariales

Fuente: Adaptado de (SPAN Systems Corporation, 2014, pág. 7)

La integración tradicional busca la interoperabilidad entre aplicaciones, emplea la topología punto a punto, la conectividad es básica y hace uso de mensajes. La integración de aplicaciones empresariales usa la topología bróker, su objetivo es la flexibilidad ya que maneja un mayor número de conexiones. La integración de servicios permite la integración y coreografía de servicios mediante un bus, persigue la reusabilidad mediante conexiones flexibles utilizando interfaces basadas en estándares.

2.1.2. Servicios web

Para Sommerville (2011) el desarrollo de la web en los años 90 transformó el intercambio de información organizacional, las PCs clientes podían acceder a información en servidores localizados fuera de su organización, este acceso era exclusivamente vía navegador web; no existía la posibilidad de que la información fuera accedida desde otros programas. Los servicios web se proponen como una alternativa para resolver dicha problemática, a través de ellos una organización hace accesible la información seleccionada mediante la definición y publicación de una interfaz para el servicio web, la interfaz define los datos disponibles así como la forma de acceder a ellos.

Según el Consorcio World Wide Web (W3C) no existe una única definición de servicio web, lo que indica cuan complejo resulta especificarla de manera que describa lo que es e involucra. Para el W3C (s.f.) un servicio web es:

Conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web.

Un servicio web, tal como se muestra en la Figura 11, posibilita la comunicación e interoperabilidad entre máquinas conectadas a la red, la comunicación se lleva a cabo a través del envío de mensajes entre cliente y servidor. El cliente realiza una solicitud al servidor, éste genera y envía una respuesta. Para Kaviya y Selvakumar (2015) un servicio web es un módulo de software que tiene una funcionalidad determinada y se invoca a través de la red; es independiente de la plataforma, puede ser escrito en cualquier lenguaje. Sus principales involucrados son proveedor, consumidor y registro.

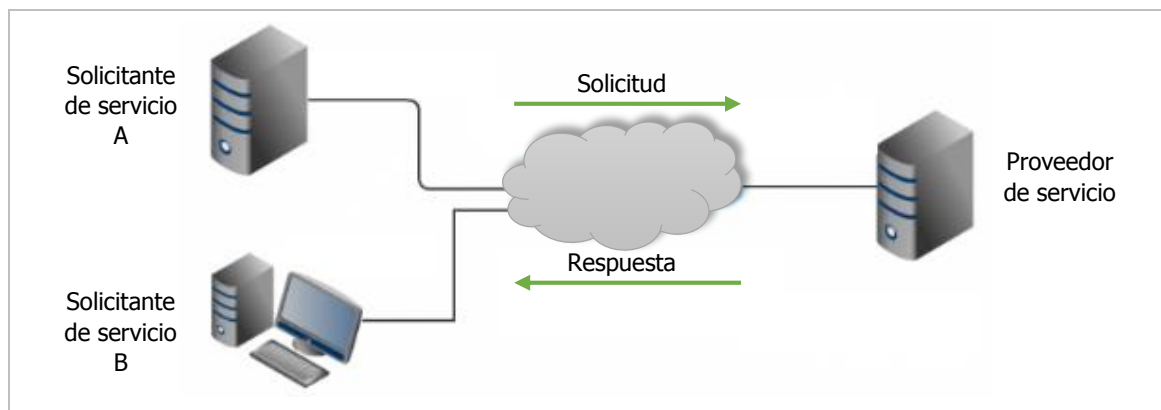


Figura 11. Servicio web

Fuente: Adaptado de (Srinivasan , 2014)

Según el Departamento de Ciencia de la Computación e Inteligencia Artificial de la Universidad de Alicante (2014) las características esperadas de un servicio web son: *acceso a través de la web* mediante la utilización de estándares para transporte y formateo de mensajes; *auto-descripción* un servicio web debe proporcionar la descripción de sí mismo para que pueda ser utilizado por cualquier aplicación, y *localización* para que un servicio web con una determinada funcionalidad sea localizado por las aplicaciones que lo requieran.

OASIS (2010) en el modelo de referencia para la arquitectura orientada a servicios 1.0 explica el funcionamiento de un servicio:

- Un servicio posibilita el acceso a una o varias funcionalidades a través de una interfaz definida, considerando las especificaciones establecidas en su descripción.
- El proveedor es la entidad que proporciona el servicio, los consumidores no siempre pueden ser conocidos por éste y el uso que le den al servicio puede ir más allá de lo inicialmente concebido por el proveedor.
- La interfaz de acceso al servicio especifica cómo se accede a las funcionalidades, que pueden ser implementadas por uno o más procesos inclusive mediante la invocación a otros servicios.

- La implementación del servicio está oculta para el consumidor, la interfaz expone la información necesaria para que el servicio pueda ser invocado por éste.
- La invocación de un servicio puede tener dos resultados: información devuelta o el cambio de estado compartido por entidades establecidas.

Los servicios web pueden ser utilizados para “integrar aplicaciones escritas en diferentes lenguajes y que se ejecutan en plataformas diferentes” (Departamento de Ciencia de la Computación e Inteligencia Artificial - Universidad de Alicante, 2014, pág. 2). Gracias a los servicios web una organización puede integrar sus aplicaciones, independientemente de lenguajes de implementación, tecnologías y plataformas de ejecución; permitiendo que las aplicaciones intercambien información y la presenten de manera uniforme y coherente (Dirección General de Tecnologías de la Información - Generalitat Valenciana, 2016).

Existen dos clases de servicios web: basados en SOAP (Simple Object Access Protocol) y en REST (Representational State Transfer). Diferentes tipos de aplicaciones pueden implementarse mediante servicios web basados en SOAP o RESTful (Mumbaikar y Padiya, 2013). “SOAP es un estándar del W3C, permite el intercambio de información en un entorno distribuido, es confiable, seguro y soporta transacciones. REST no es un estándar ni está promovido por algún proveedor, es un estilo arquitectónico para sistemas hipermedia distribuidos” (Dudhe y Sherekar, 2014, pág. 1).

2.1.2.1. Servicios web basados en SOAP

Son denominados servicios web “big”, hacen uso de mensajes XML según el estándar SOAP para comunicarse entre sí; utilizan WSDL para describir los servicios web (Departamento de Ciencia de la Computación e Inteligencia Artificial - Universidad de Alicante, 2014). La Figura 12 presenta de manera general el funcionamiento de un servicio web basado en SOAP, en donde los clientes realizan solicitudes a través de mensajes SOAP en HTTP, y los servidores generan respuestas en formato XML.

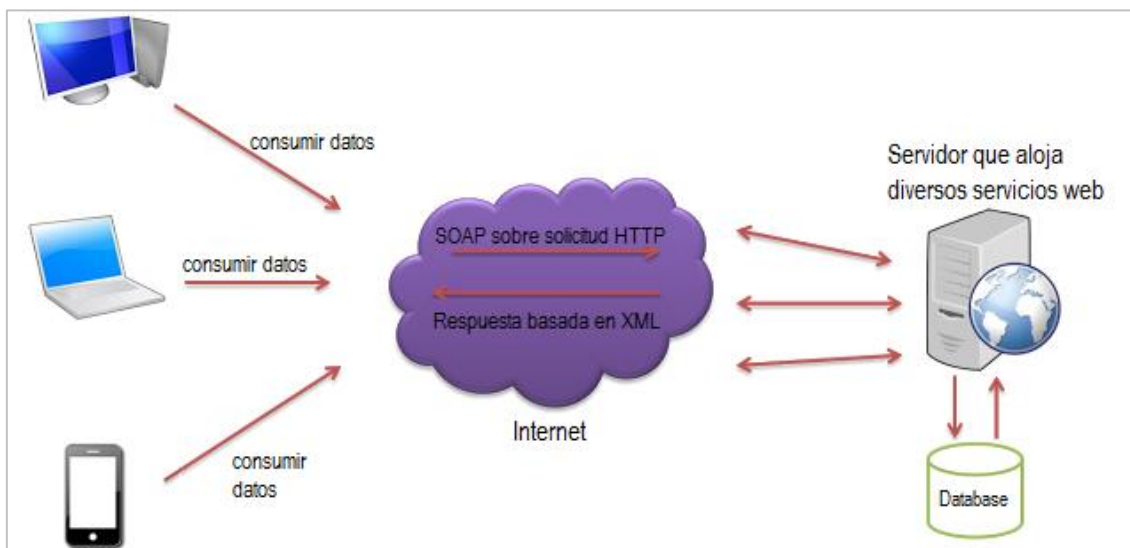


Figura 12. Servicio web basado en SOAP

Fuente: Adaptado de (Chalam, 2012)

Un servicio web basado en SOAP está caracterizado por los siguientes estándares:

- **SOAP** (Simple Object Access Protocol).- Es un protocolo basado en XML que permite el “intercambio de información en entornos descentralizados y distribuidos”. Remitente y destinatario interactúan mediante el intercambio de mensajes SOAP compuestos de tres elementos: sobre, reglas de codificación y estilos de comunicación. El sobre describe el contenido del mensaje y cómo debe ser procesado, constituye el elemento de nivel superior del documento XML, incluye cero o más cabeceras y un cuerpo; las cabeceras contienen información de control, el cuerpo la identificación del mensaje y sus parámetros. Las reglas de codificación especifican las instancias de tipos de datos establecidas por la aplicación y los mecanismos de serialización para intercambiarlas. Los estilos de comunicación pueden seguir el formato de llamada a procedimiento remoto o el formato orientado a mensajes (IBM Knowledge Center, 2017).
- **WSDL** (Web Services Description Language).- Se basa en XML, es un lenguaje empleado para efectuar la descripción de servicios web. Mediante WSDL los proveedores especifican cómo deben ser invocados los servicios web, cuáles con sus parámetros y datos retornados. En la descripción de un servicio web se

especifican sus elementos a través de etiquetas XML, dichos elementos pueden ser abstractos o concretos. Son abstractos al momento de definirlos es decir cuando se establece las operaciones, parámetros, respuestas, mensajes de error con sus respectivas definiciones de tipos. Los elementos concretos especifican el cómo y dónde del servicio, es decir cómo se invoca al servicio, qué protocolo se emplea, dónde se encuentra disponible (Departamento de Ciencia de la Computación e Inteligencia Artificial - Universidad de Alicante, 2014).

- **UDDI** (Universal Description Discovery and Integration).- Es la especificación de descripción, descubrimiento e integración universal que indica cómo se publican y descubren los servicios web. En un directorio distribuido se almacena información en formato XML sobre las organizaciones, su clasificación y servicios web ofrecidos (Departamento de Ciencia de la Computación e Inteligencia Artificial - Universidad de Alicante, 2014).

Un proveedor de servicio lo describe a través de un archivo WSDL, envía dicha información junto con otra en un mensaje SOAP para realizar el registro en UDDI de manera que el servicio sea localizable. Cuando un cliente requiere un servicio realiza la búsqueda en UDDI, en este proceso se emplean mensajes SOAP tanto para el envío del requerimiento por parte del cliente como para la recepción de la respuesta generada por UDDI.

2.1.2.1.1. Características

Según Patil, Kshirsagar y Jaypal (2012) un servicio web es un tipo de servicio que presenta las siguientes características:

- **Basado en XML.**- Los servicios web hacen uso de XML tanto en la representación de datos como en las capas de transporte de datos; el uso de XML hace que sean altamente interoperables.
- **Débilmente acoplado.**- En una arquitectura fuertemente acoplada el cliente, la lógica del negocio y servidor están estrechamente vinculados, de manera que si

cambia una interfaz las demás también tendrán que ser modificadas. En una arquitectura débilmente acoplada, múltiples cambios en unas interfaces no afecta a las demás.

- **Descubrimiento Automático.**- El proveedor publica la descripción del servicio en un registro utilizando UDDI. Los consumidores buscan en el registro la especificación para un determinado servicio. Servicios de diferentes proveedores podrían calzar con dicha especificación, el cliente se vincula a un proveedor de forma dinámica.
- **Auto-descripción.**- Mediante WSDL se especifica la descripción funcional del servicio, incluye las interfaces que permiten a los clientes invocar al servicio web.
- **Agnóstico de la plataforma.**- Los servicios se pueden implementar en diversas plataformas y tecnologías; son independientes de la red, sistema operativo y plataforma debido al uso de XML.
- **Capacidad para ser síncrono o asíncrono.**- Un servicio web puede ser invocado en modo síncrono o asíncrono; en el primer caso se procesa una solicitud a la vez, por tanto una nueva solicitud tiene que esperar hasta que la anterior finalice. En modo asíncrono varios solicitantes pueden invocar el servicio al mismo tiempo.
- **Soporte a llamada de procedimiento remoto.**- Los servicios web posibilitan a los clientes invocar procedimientos en objetos remotos mediante un protocolo basado en XML.
- **Basado en estándares abiertos.**- Los servicios web se basan en estándares como XML, HTTP y otros proyectos de código abierto.

Los servicios web exponen su funcionalidad a través de la red, pueden ser invocados de forma remota a través de HTTP, permiten que varias aplicaciones se comuniquen entre sí y puedan compartir información, en la comunicación hacen uso de protocolos estándares apoyados por la industria (Tutorials Point, 2015).

2.1.2.1.2. Arquitectura

La arquitectura de un servicio web puede ser descrita desde dos puntos de vista (Tutorials Point, 2015):

Considerando las funciones de los actores del servicio web.- En este caso, como se muestra en la Figura 13 participan proveedor, solicitante y registro. El proveedor es quien implementa el servicio y lo publica en Internet. El solicitante consume el servicio web, al invocarlo abre una conexión de red y envía una solicitud XML. El registro es un directorio centralizado de servicios, proporciona un lugar central donde los desarrolladores publican nuevos servicios y encuentran los existentes.

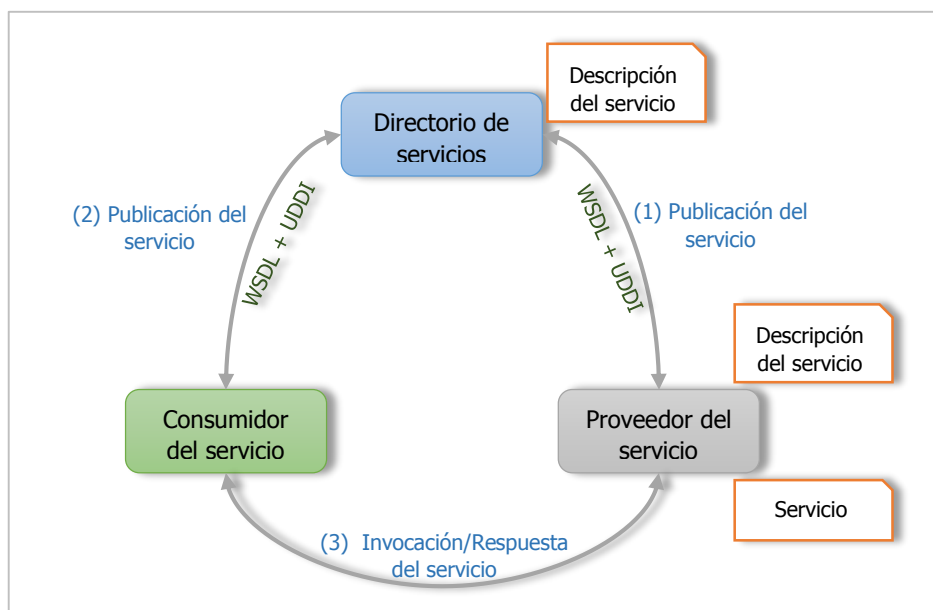


Figura 13: Roles en un servicio web

Fuente: Adaptado de (W3C Working Draft, 2002)

En base a la pila de protocolos del servicio web (Figura 14).

- **Servicio de transporte.-** Esta capa es responsable de transportar mensajes entre aplicaciones, incluye el protocolo de transporte de hipertexto (HTTP), protocolo de transferencia de correo simple (SMTP), protocolo de transferencia de archivos (FTP) y protocolos como Blocks Extensible Exchange Protocol (BEEP).

- **Mensajería XML.**- Esta capa es responsable de codificar los mensajes en formato XML para que puedan ser entendidos en cada extremo. Actualmente incluye XML-RPC y SOAP.
- **Descripción del servicio.**- Capa encargada de describir la interfaz pública de un servicio web a través de WSDL.
- **Descubrimiento del servicio.**- Esta capa tiene como función la centralización de servicios en un registro común facilitando la publicación y búsqueda. El descubrimiento de servicios se realiza a través de UDDI.

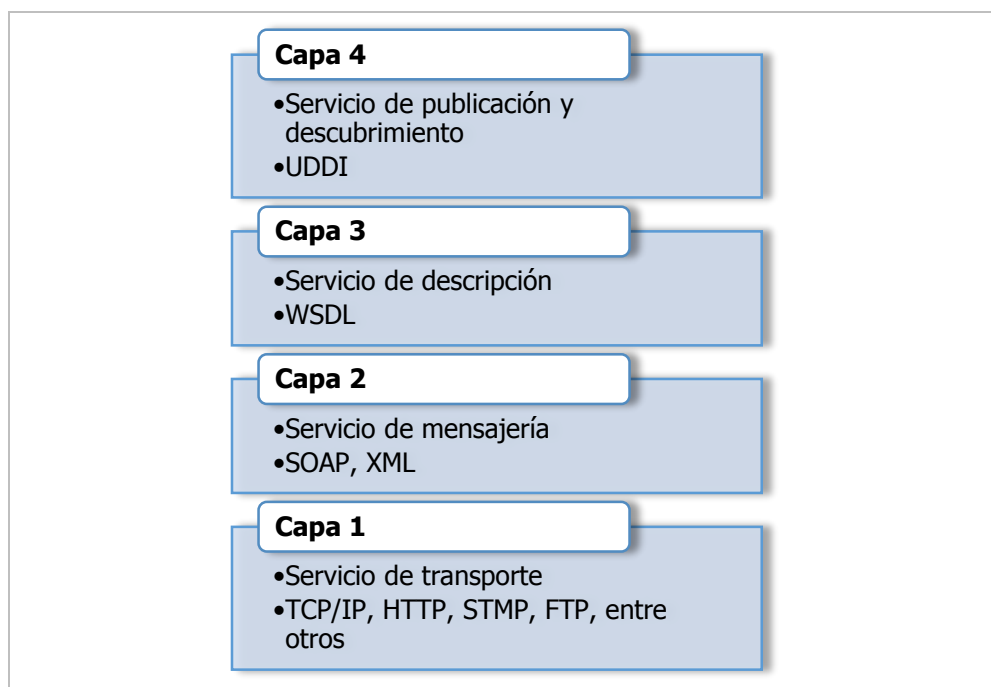


Figura 14: Pila de protocolos de un servicio web

Fuente: Adaptado de (SERVICIOS WEB ALICANTE - dwalicante, 2016)

Conforme los servicios web evolucionan, se pueden agregar capas adicionales y tecnologías a cada capa. Los servicios web basados en SOAP han sido la opción para muchas empresas siendo ampliamente utilizados, sin embargo durante la última década los servicios web basados en REST están ganando espacio, a continuación se describen con mayor detalle.

2.1.2.2. Servicios REST

REST (REpresentational State Transfer - Transferencia de Estado Representacional) es un estilo arquitectónico que define a la web “como una aplicación hipertexto distribuida, cuyos recursos vinculados se comunican intercambiando representaciones de estado del recurso” (pág. 1); los recursos constituyen los elementos básicos de la plataforma web, al trabajar con REST una de las primeras tareas radica en su identificación así como en la caracterización de las relaciones entre recursos (Kumar, 2015).

REST no es un estándar, más bien establece los principios arquitectónicos para construir aplicaciones o servicios web, es por ello que no está vinculado con una determinada tecnología. Los servicios web implementados según la arquitectura REST se denominan API RESTful o API REST.

Las características de REST son (Fredrich, 2017):

- **Interfaz uniforme.**- Define la interfaz entre clientes y servidores, la arquitectura es simple y desacoplada lo que permite la evolución independiente de cada componente. Los principios de la interfaz uniforme son:
 - *Basada en recursos:* Un elemento fundamental de REST es el recurso, identificado por un Identificador de Recursos Uniforme (URI - Uniform Resource Identifier) y representado por un documento. REST realiza operaciones específicas sobre los recursos identificados por una URI. Tanto las solicitudes de clientes como respuestas de servidores consisten en la transferencia de representaciones de recursos.
 - *Manipulación de recursos en base a su representación:* La representación de un recurso con los respectivos metadatos permite actualizarlo o eliminarlo, siempre y cuando se cuente con los permisos correspondientes.
 - *Mensajes auto-descriptivos:* Cada mensaje contiene la información suficiente que indica cómo debe ser procesado.
 - *HATEOAS:* Hypermedia as the Engine of Application State - Hipertexto como el motor del estado de aplicación. Clientes y servidores entregan el estado de la solicitud como hipertexto.

- **Sin estado.**- En REST no hay necesidad de almacenar el estado de las peticiones en el servidor o cliente, dentro del mensaje debe incluirse la información necesaria para manejarlas; es decir el estado que permite gestionar una petición está contenido en ella misma.
- **Cacheable.**- Los clientes pueden almacenar las respuestas recibidas en caché, esta opción con una adecuada gestión permite optimizar el tráfico de red, pues se reduce el número de interacciones entre clientes y servidores. Las respuestas -de manera implícita o explícita- deben ser definidas como “cacheables” para evitar que los clientes hagan uso de datos obsoletos o que no correspondan a una solicitud.
- **Cliente/servidor.**- La interfaz uniforme separa a clientes de servidores. REST es una arquitectura en donde cliente y servidor pueden desarrollarse de forma independiente, inclusive pueden ser modificados o reemplazados mientras que la interfaz no sea alterada.
- **Sistema a capas.**- REST se caracteriza por ser una arquitectura jerárquica de componentes, cada capa implementa una determinada funcionalidad. Por ejemplo puede incluirse un servidor intermedio para gestionar el equilibrio de carga, gestión de caché o políticas de seguridad.

Corporaciones como Facebook, Google, Twitter, Youtube, entre otras, proporcionan APIs basadas en REST para brindar acceso a sus funcionalidades, de manera que puedan ser incorporadas en aplicaciones de terceros. Por ejemplo, los servicios web de Google Maps API dan acceso a datos geográficos que pueden utilizarse en aplicaciones de mapas.

2.1.2.2.1. Operaciones empleadas en REST

“REST se basa en un conjunto de operaciones de transferencia que son universales para cualquier sistema de almacenamiento y recuperación de datos” (Markey y Clynch, 2013, pág. 3). Las aplicaciones RESTful hacen uso de peticiones HTTP para efectuar operaciones CRUD (Create, Read, Update y Delete) con la finalidad de crear, leer, modificar y eliminar datos respectivamente (Kumar, 2015), la Tabla 1 presenta el nombre de cada operación con su respectiva descripción.

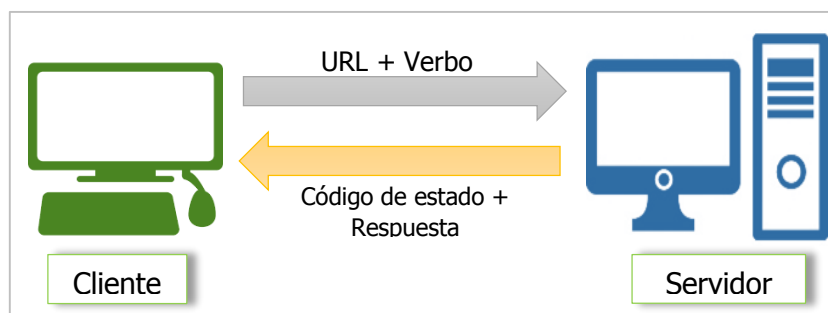
Tabla 1. Operaciones CRUD HTTP

Operación CRUD	Operación en REST	Descripción
Create (Crear)	POST	Crea un nuevo recurso
Retrieve (Leer)	GET	Recupera el estado actual del recurso
Update (Actualizar)	PUT	Modifica un recurso
Delete (Borrar)	DELETE	Borra un recurso

Fuente: (Sinha, Khatkar y Gupta, 2014, pág. 387)

En un sistema RESTful los recursos pueden representar cualquier cosa del mundo real, sin embargo las operaciones que un cliente puede realizar con ellos son limitadas; clientes y servidores interactúan a través del envío de mensajes según un protocolo determinado, que en el caso de la web es HTTP (Richardson, Amundsen y Ruby, 2013). Los métodos HTTP son también conocidos como verbos. Los métodos GET, POST, DELETE y PUT son los comúnmente más utilizados, sin embargo existen otros.

La Figura 15 muestra la interacción entre cliente y servidor en un sistema RESTful. El cliente formula una petición involucrando la URL del recurso más el verbo que indica la acción que se desea ejecutar sobre él; en tanto que el servidor procesa la petición y como parte de la respuesta devuelta incluye un código de estado HTTP que informa al cliente sobre el estado de su solicitud.

**Figura 15.** Interacción cliente y servidor en RESTful

Fuente: Adaptado de (Kumar, 2015, pág. 4)

Un código de estado HTTP está conformado por tres dígitos, es un valor que se adjunta a la respuesta HTTP con el objetivo de informar al cliente sobre el resultado de su petición. En la web cuando existen errores asociados a una respuesta, lo que usualmente se hace es leer el mensaje relacionado al error para corregirlo sin prestar atención al código de estado; en cambio para las API los códigos de estado HTTP son importantes ya que indican al cliente cómo debe tratar el contenido del recurso, o qué hacer si no puede entenderlo (Richardson, Amundsen y Ruby, 2013).

El primer dígito del código de estado corresponde a la categoría, los dos siguientes no especifican una función en particular. El RFC 7231 describe la semántica y contenido del protocolo de transferencia de hipertexto (Hypertext Transfer Protocol HTTP/1.1), en una de las secciones se incluye los códigos de estado HTTP (IETF, 2014) organizados por categorías. A continuación se explican los códigos de estado más representativos para cada categoría (RestPatterns, 2017):

- **Categoría 1XX (Informativo).**- Este grupo indica que la solicitud fue recibida y el proceso continúa, es utilizado para mostrar el estado de las negociaciones entre clientes y servidores. Los principales códigos de estado HTTP de esta categoría se describen en la Tabla 2.

Tabla 2. *Principales códigos de estado HTTP – Categoría 1XX*

Código	Descripción
100 Continuar	Indica al cliente que debe continuar con su solicitud ya que la parte inicial de la misma fue recibida por el servidor y todavía no ha sido rechazada, el servidor enviará una respuesta cuando la solicitud se haya enviado.

Fuente: Elaboración propia.

- **Categoría 2XX (Exitoso).**- Los códigos de esta categoría muestran que la solicitud fue recibida y procesada con éxito. La Tabla 3 contiene los principales códigos de estado HTTP de esta categoría.

Tabla 3. Principales códigos de estado HTTP – Categoría 2XX

Código	Descripción
200 OK	Solicitud exitosa. La información adicional devuelta depende de la naturaleza de la solicitud.
201 Creado	La solicitud fue realizada, dando como resultado la creación de un recurso. Como parte de la respuesta devuelta se incluye la URL del recurso. Si el servidor creó el recurso debe devolver este código, si la creación no se realiza todavía entonces devuelve el código 202.
202 Aceptado	Indica que la solicitud ha sido aceptada para su posterior procesamiento. Una vez que la tarea se lleve a cabo podría o no ser rechazada.
204 Sin contenido	Indica que el servidor ha cumplido la solicitud de forma satisfactoria pero no requiere devolver información adicional. El objetivo de este código es evitar que haya un cambio en la vista del documento del cliente.

Fuente: Elaboración propia.

- **Categoría 3XX (Redirección).**- Esta categoría indica que deben realizarse acciones adicionales para que la solicitud se complete, por ejemplo la redirección hacia otra página. Los principales códigos de estado HTTP de la categoría 3XX se describen en la Tabla 4.

Tabla 4. Principales códigos de estado HTTP – Categoría 3XX

Código	Descripción
301 Movido permanentemente	El recurso solicitado cambió de URI de forma permanente por lo tanto cualquier referencia posterior debe utilizar los URIs devueltos; que usualmente el servidor retorna en el campo Location, además de una nota de hipertexto con un hipervínculo a la nueva URI.
304 No modificado	El cuerpo de la respuesta no tiene contenido, se utiliza cuando existen datos para enviar al cliente, pero éste ya los tiene. El código 304 se emplea con solicitudes condicionales HTTP; en el caso de que se solicite un documento con el encabezado If-Modified-Since mas una fecha, el servidor devuelve 304 si el documento no fue modificado desde entonces.

Fuente: Elaboración propia.

- **Categoría 4XX (Error del cliente).**- La solicitud formulada por el cliente tiene sintaxis incorrecta o no puede ser realizada. El servidor debería incluir en su respuesta la explicación del error así como información sobre si esa condición es temporal o permanente. Los principales códigos de estado HTTP para esta categoría se describen en la Tabla 5.

Tabla 5. Principales códigos de estado HTTP – Categoría 4XX

Código	Descripción
400 Petición incorrecta	La solicitud enviada por el cliente no puede ser entendida por el servidor debido a problemas de sintaxis.
401 No autorizado	La solicitud requiere autenticación del cliente. El error puede darse porque no se proporcionó una credencial o la suministrada es incorrecta.
403 Prohibido	La solicitud es entendida pero el servidor no quiere efectuarla, este caso aplica cuando los recursos están accesibles durante ciertos periodos o desde determinadas IP. El servidor puede o no dar a conocer la razón del error, en el segundo caso retorna el código 404.
404 No encontrado	El servidor no encuentra un recurso que coincida con la URI de la solicitud. Código empleado cuando el servidor no desea dar a conocer por qué rechaza la solicitud o no es posible utilizar otra respuesta.
409 Conflicto	Un conflicto con el estado del recurso causa que la solicitud no pueda ser completada. Este código debería ser empleado cuando se espera que el cliente resuelva el conflicto, en ese caso el servidor debería proporcionar la información suficiente para tal efecto.

Fuente: Elaboración propia.

- **Categoría 5XX (Error del servidor).**- El servidor no pudo cumplir con una solicitud que aparentemente es válida, los principales códigos de estado HTTP de esta categoría se describen en la Tabla 6.

Tabla 6. Principales códigos de estado HTTP – Categoría 5XX

Código	Descripción
500 Error interno de servidor	Indica que ha ocurrido una excepción que impide que la solicitud pueda ser cumplida.
503 Servicio no disponible	Indica que temporalmente el servidor no puede atender la solicitud, debido a sobrecarga o por encontrarse en mantenimiento.

Fuente: Elaboración propia.

Los verbos HTTP utilizados por REST para operaciones CRUD (Richardson, Amundsen y Ruby, 2013) son:

- **GET.**- Método utilizado para obtener la representación de un recurso identificado por una URL, es decir permite consultar su estado. Este método es seguro pues implica únicamente la solicitud de información, enviar una solicitud GET es igual a no enviarla, no existen efectos en el estado de un recurso. El código de respuesta

para una solicitud GET comúnmente es 200, aunque el código 301 también puede ser devuelto.

- **DELETE.**- Método que permite eliminar un recurso, un cliente solicita la acción esperando que el servidor lo destruya de manera que en adelante ya no sea posible referenciarlo. Los códigos de estado devueltos son 204 (recurso eliminado), 200 (eliminación efectuada) o 202 (solicitud aceptada). Si un cliente intenta acceder a un recurso eliminado el servidor devolverá un código de estado 404 (no encontrado). DELETE no es un método seguro, enviar una solicitud DELETE no es igual a no enviarla. DELETE es idempotente (cuando una operación se ejecuta una o más veces y produce el mismo resultado), esta característica es útil en el caso de que se presenten problemas de conexión impidiendo que un servidor responda, entonces el cliente al no conocer que pasó con su petición volverá a enviarla hasta obtener una respuesta.
- **POST.**- Método empleado usualmente para la creación de un recurso. POST se utiliza con dos propósitos, el primero consiste en la creación de un recurso por debajo de otro (POST to append); en ese caso el cliente envía una representación del recurso que desea crear en el cuerpo de la solicitud, los códigos de solicitud devueltos con frecuencia son 201 (recurso creado) o 202 (aceptado). POST no es seguro ni idempotente, no solo permite crear nuevos recursos sino también encapsular otras tareas (POST sobrecargado), puede ser utilizado por ejemplo para realizar una operación de modificación sobre los datos de un formulario HTML, equivalente a una operación PUT pero HTML no puede activarla por tanto utiliza POST. Este comportamiento de POST es válido debido a que en la especificación de HTTP las tareas de POST son “proporcionar un bloque de datos, ... publicación de mensajes, ... crear un nuevo recurso; ... y adicionar datos a las representaciones existentes de un recurso” (IETF, 2014, pág. 26) ; entonces “la función real realizada por el método POST es determinada por el servidor y suele depender de la URI de solicitud” (RestPatterns, 2017).

- **PUT.**- Se emplea para editar un recurso. El cliente modifica la representación del recurso (respuesta de una solicitud GET) y la devuelve como “carga útil” para una solicitud PUT. El servidor podría rechazar una solicitud PUT por inconsistencias en el cuerpo de la entidad o por otros motivos; en el caso de que la acepte, cambia el estado del recurso y devuelve un código 200 o 204. PUT es idempotente, también puede ser utilizado para crear un recurso si se conoce su URL.

Según Kumar (2015) REST es más flexible en comparación con SOAP, soporta distintos formatos además de XML, los mensajes de REST consumen menor ancho de banda debido a que son más pequeños, presenta mayor rendimiento, menor acoplamiento entre clientes REST y servidores, el tiempo de aprendizaje es menor, resulta ser una buena opción cuando se requiera desarrollar APIs ligeras que soporten operaciones CRUD y que se caractericen por un buen rendimiento.

2.1.3. WordPress

Un sistema de gestión de contenido (CMS - Content Management System) ofrece herramientas para la administración de cualquier tipo de contenido estructurado como noticias, imágenes, videos, entre otros. Sitios web de diferente índole están construidos con CMS, en dicha tarea pueden intervenir varios usuarios con distintos roles (por ejemplo: administrador, editor, autor, colaborador, suscriptor).

“WordPress es un sistema de gestión de contenidos de libre distribución, compatible con estándares, rápido, ligero y gratuito, con configuraciones y características predeterminadas y un núcleo muy personalizable” (WordPress Codex, 2017). En la documentación oficial de Wordpress destacan características como la facilidad de uso, CMS probado con amplia trayectoria, cuenta con el respaldo de una extendida comunidad de desarrolladores, es multi-sitio, proporciona un conjunto de potentes funcionalidades centradas en el contenido, posibilita la creación de blogs y sitios web de tipo empresarial; garantiza a los usuarios la libertad para usarlo, estudiarlo, modificarlo y redistribuirlo; entre otras. La Figura 16 presenta el diagrama de la base de datos de WordPress, contiene las tablas con sus respectivos campos, así como las relaciones entre ellas.

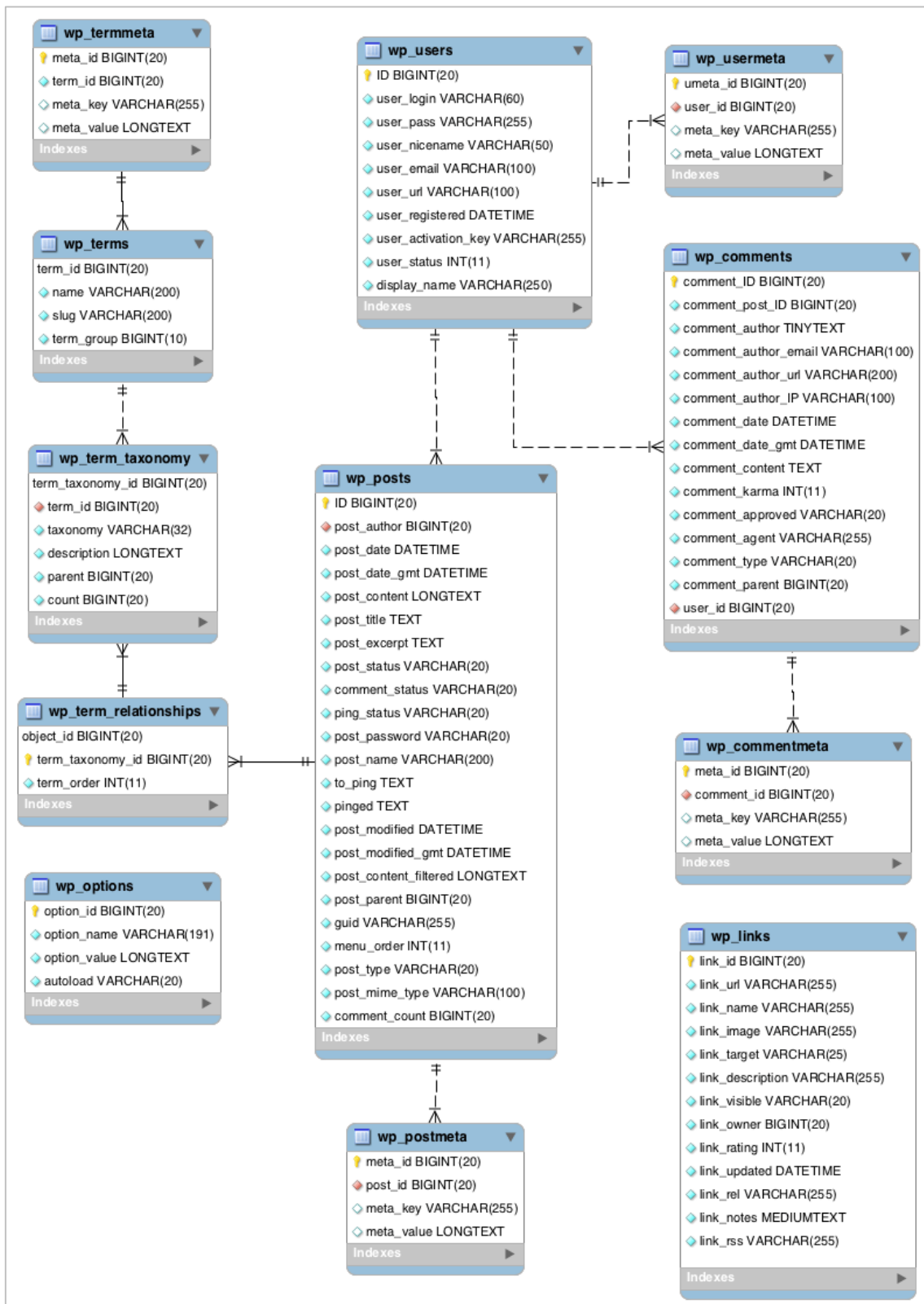


Figura 16: Diagrama de la base de datos de WordPress

Fuente: (WordPress Codex, 2017)

Conocer la estructura de la base de datos de WordPress es de utilidad especialmente cuando se van a desarrollar módulos o a trabajar directamente con su información, en la Tabla 7 se describe de manera general el propósito de cada tabla de la base de datos.

Tabla 7. Estructura de la base de datos de WordPress

Nombre de la tabla	Descripción
_commentmeta	Los metadatos de cada comentario se almacenan en esta tabla.
_comments	Contiene los comentarios de WordPress.
_links	Los enlaces agregados mediante la opción <i>Enlaces</i> se conservan en esta tabla.
_options	Las opciones que se definen mediante <i>Administración</i> → <i>Configuración</i> son guardadas esta tabla.
_postmeta	La metadata de cada post se almacena en _postmeta, varios plugins también pueden guardar información adicional en esta tabla.
_posts	Los posts son la parte medular de los datos. Esta tabla almacena posts (publicaciones) de WordPress, además de páginas e ítems del menú de navegación.
_terms	La tabla _terms contiene las categorías de las entradas y enlaces, y las etiquetas de las entradas.
_term_relationships	La asociación de entradas y enlaces con categorías y/o etiquetas se almacenan en la tabla _terms_relationships.
_term_taxonomy	Las taxonomías (categorías y etiquetas) en WordPress permiten agrupar contenido. En la tabla _term_taxonomy se guarda la taxonomía de cada elemento de la tabla _terms.
_usermeta	Los metadatos relacionados con cada usuario se almacenan en la tabla _usermeta.
_users	Contiene el listado de usuarios.

Fuente: (WordPress Codex, 2017)

Una de las principales tablas de WordPress es *_posts*, su nombre podría indicar que almacena exclusivamente publicaciones (denominadas también artículos o posts) del blog, sin embargo también contiene páginas, archivos adjuntos, elementos de menús y otra información concerniente con la personalización del sitio. Los campos de la tabla *_posts* son (Poulson, 2016):

- **ID.**- Identificador único de cada publicación.
- **post_author.**- Identificador del usuario creador de la publicación (campo relacionado con la tabla `_users`).
- **post_date.**- Hora y fecha de creación de la publicación.
- **post_date_gmt.**- Almacena la hora y fecha de creación de la publicación en formato GMT, mediante éste se evita la dependencia de la zona horaria de un sitio.
- **post_content.**- Guarda el contenido de la publicación.
- **post_title.**- Título de la publicación.
- **post_excerpt.**- Contiene la introducción personalizada o resumen del contenido.
- **post_status.**- Guarda el estado de la publicación (*draft*/borrador, *pending*/pendiente, *private*/privado, *publish*/publicado).
- **comment_status.**- Indica si se permiten comentarios.
- **ping_status.**- Especifica si el post permite ping y trackbacks.
- **post_password.**- Contiene la contraseña opcional para visualizar la publicación.
- **post_name.**- URL amigable (slug) del título de la publicación.
- **to_ping.**- Listado de URLs de WordPress a las que se debería enviar pingbacks cuando se actualiza la publicación. Cuando un post del sitio web A referencia a un post del sitio web B, entonces desde B se envía un pingback notificando a A, éste a su vez verifica el pingback indicando a B que el enlace al post es correcto.
- **pinged.**- Listado de URLs de WordPress a las que se ha enviado pingbacks cuando se actualiza la publicación.
- **post_modified.**- Almacena hora y fecha correspondiente a la última modificación.
- **post_modified_gmt.**- Es la hora y fecha GMT de la última modificación.
- **post_content_filtered.**- Empleado por plugins para almacenar en caché una versión de `post_content`.
- **post_parent.**- Este campo es empleado para establecer la relación entre dos publicaciones, cuando una de ellas por ejemplo es una revisión, archivo adjunto u otro tipo.

- **guid.**- Identificador único global, es la URL de la publicación (no la versión de permalink).
- **menu_order.**- Almacena el número de visualizaciones de páginas y otros tipos de entradas.
- **post_type.**- Indica el tipo de contenido (post, page, attachment y nav_menu_item).
- **post_mime_type.**- Utilizado para los archivos adjuntos, especifica el tipo MIME del archivo.
- **comment_count.**- Número total de comentarios, pingbacks y trackbacks. Un trackback es similar a un pingback pero funciona manualmente. Pingbacks y trackbacks permiten establecer comunicación entre sitios web.

Un administrador de WordPress no requiere conocer el lenguaje de script PHP tampoco el gestor de base de datos MySQL para utilizarlo; sin embargo, entender cómo y dónde se almacena la información en WordPress ayuda a comprender su funcionamiento, pudiendo ser de utilidad en la resolución de problemas que pudieran presentarse con el CMS.

2.1.3.1. API REST de WordPress

La API REST de WordPress (WP) permite a los desarrolladores realizar operaciones CRUD sobre el contenido de WordPress de manera remota, utilizando JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) como formato de solicitud/respuesta; esta característica hace posible que para el procesamiento sea posible utilizar cualquier lenguaje de programación capaz de realizar peticiones HTTP e interpretar JSON, construyendo o empleando aplicaciones ya existentes que accedan al contenido de un sitio web (REST API Handbook, 2017).

La API REST WP provee un conjunto de endpoints para los tipos de contenido de WordPress... Los endpoints son funciones que se pueden utilizar con la API para leer y escribir datos de contenido. Mediante el uso de funciones de endpoint con parámetros se envía la respuesta al cliente. (PressTigers, 2017)

Mediante la API REST WP es posible acceder a datos públicos y privados, en el primer caso el acceso se realiza de forma anónima, mientras que para el segundo se requiere autenticación solo entonces gran parte de las tareas de administración de contenido estarán disponibles.

Los conceptos clave relacionados con la API REST WP son (REST API Handbook, 2017):

- **Rutas y endpoints.**- Una ruta es un identificador único de recursos (URI) que puede ser asignado a distintos métodos HTTP. Un endpoint consiste en la asignación de un método HTTP específico a una ruta.
- **Solicitudes.**- La clase WP_REST_Request es empleada para almacenar y recuperar información de una determinada solicitud. Las solicitudes pueden enviarse de manera remota o local, en el primer caso a través de HTTP y en el segundo mediante PHP con WordPress. Cuando se realiza una solicitud automáticamente se generan objetos de la clase WP_REST_Request, la respuesta obtenida dependerá de la solicitud realizada.
- **Respuestas.**- Son los datos obtenidos de la API que pueden consistir en datos deseados o a su vez errores, WP_REST_Response permite interactuar con los datos devueltos por los endpoints.
- **Esquema.**- El esquema de la API contiene la definición de su estructura, es decir propiedades que devuelve y parámetros que puede aceptar; mediante el esquema se puede validar las solicitudes que se realizan a la API.
- **Clases de controlador.**- Permiten unificar los elementos de la API para que trabajen en conjunto; mediante una clase de controlador es posible registrar rutas y endpoints, gestionar solicitudes, hacer uso de esquemas y crear respuestas.

La referencia para desarrolladores de la API REST de WP contiene la información de los endpoints disponibles, parámetros y formatos de datos de respuesta; para cada recurso incluye una descripción completa del esquema, así como de las operaciones para listar, recuperar, crear, modificar y borrar. La Tabla 8 presenta las rutas base para los recursos de WordPress.

Tabla 8. Referencia para desarrolladores de los endpoints de la API REST de WP

Recurso	Ruta base
Posts	/wp/v2/posts
Revisiones de posts	/wp/v2/revisions
Categorías	/wp/v2/categories
Etiquetas	/wp/v2/tags
Páginas	/wp/v2/pages
Comentarios	/wp/v2/comments
Taxonomías	/wp/v2/taxonomies
Medios	/wp/v2/media
Usuarios	/wp/v2/users
Tipos de posts	/wp/v2/types
Estado de posts	/wp/v2/statuses
Configuración	/wp/v2/settings

Fuente: (REST API Handbook, 2017)

Según la documentación oficial, la API REST facilita el uso de WordPress de maneras en las que solo la imaginación es el límite ya que brinda una manera “estructurada, extensible y simple para obtener datos dentro y fuera de WordPress a través de HTTP” (REST API Handbook, 2017); de forma que WordPress se convierte de ser un sistema de gestión de contenidos a un marco de aplicación con el que se puede comunicar cualquier tipo de aplicación.

2.1.4. Node.js

“Node.js es un entorno de ejecución para JavaScript que hace uso del motor de JavaScript V8 de Chrome. Node.js utiliza un modelo de E/S sin bloqueo y controlado por eventos que lo hacen liviano y eficiente” (Node.js Foundation, 2017). Node.js es multiplataforma y open source, permite desarrollar aplicaciones empleando JavaScript del lado del servidor.

La E/S asíncrona (sin bloqueo) hace posible que mientras se completan las operaciones de E/S se continúe con el procesamiento. En la programación basada en eventos el flujo del programa está determinado por eventos, que no únicamente son las acciones

realizadas por el usuario (RisingStack, 2017). Node Package Manager (npm) es el gestor de paquetes de node.js. Npm permite a los desarrolladores JavaScript reutilizar y compartir código, también gestiona las actualizaciones del código compartido.

Las características de Node.js son (Tutorials Point, 2015):

- **Asíncrono y dirigido por eventos.**- Node.js usa programación asíncrona, sin bloqueo; esto significa que cuando una tarea requiere tiempo para completarse Node.js pasa a ejecutar la siguiente, el servidor obtiene la respuesta de la tarea anterior mediante un mecanismo de notificación de eventos.
- **Rápido.**- Node.js ejecuta el código JavaScript con alta velocidad debido a que hace uso del motor V8 de Google Chrome.
- **Único hilo pero altamente escalable.**- Node.js emplea un modelo de único hilo con bucle de eventos (event loop), que posibilita al servidor responder sin bloqueos haciéndolo altamente escalable pudiendo atender una mayor cantidad de solicitudes.
- **Licencia.**- Node.js se distribuye bajo la licencia MIT.

Node ha experimentado un importante crecimiento durante los últimos años; empresas como LinkedIn y Walmart -entre otras- hacen uso de aplicaciones desarrolladas con Node.js. LinkedIn movió su backend de aplicaciones móviles desde Ruby on Rails a Node.js, la velocidad de la nueva aplicación mejoró entre 2 a 10 veces en relación a su antecesora. Walmart rediseñó su aplicación móvil con Node.js para optimizar el manejo de solicitudes concurrentes (Chrzanowska, 2017).

2.1.4.1. LoopBack

LoopBack es un framework para Node.js de la compañía Strongloop, se caracteriza por ser altamente extensible y open source; permite crear de manera fácil API's Rest dinámicas, acceder a diferentes fuentes de datos como Oracle, MySQL, PostgreSQL, MS SQL Server, MongoDB y otras; crear aplicaciones para Android, iOS y JavaScript SDK; ejecutar las aplicaciones de manera local o en la nube (StrongLoop, 2017).

No se ha establecido un framework dominante para Node.js, sin embargo LoopBack se destaca debido a que posibilita la creación de APIs, conectándolas con diferentes fuentes de datos. A partir de una definición de modelo de datos se genera de manera sencilla una API REST funcional que puede ser invocada por cualquier cliente (Jovanovic, 2015).

Un modelo es un elemento clave en LoopBack, representa a diversas fuentes de datos tales como bases de datos u otros servicios; al definir un modelo automáticamente incluye una API REST con un conjunto de operaciones CRUD (LoopBack , 2017). La Figura 11 presenta los principales módulos de LoopBack, con sus relaciones y dependencias.

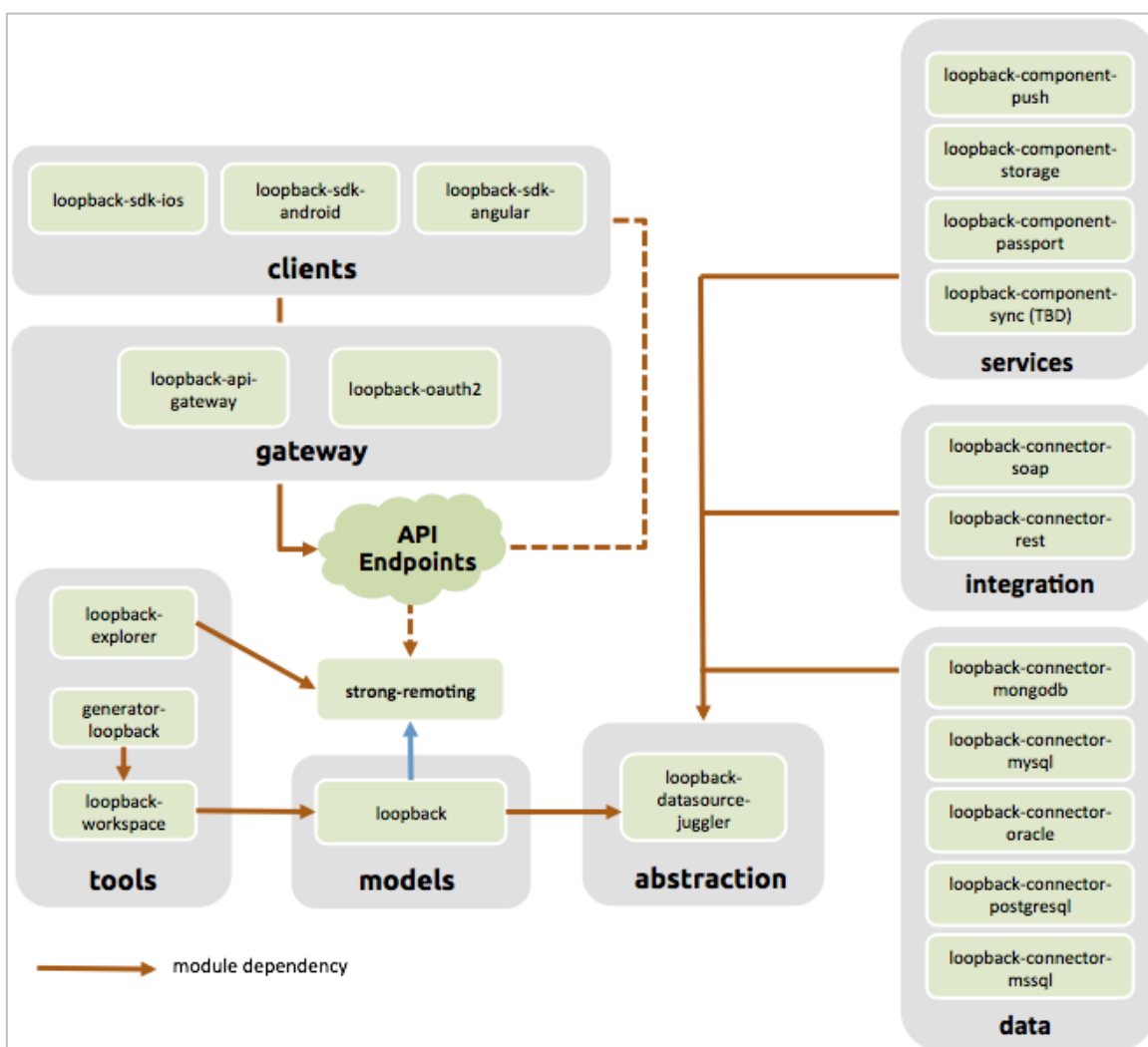


Figura 17: Módulos de LoopBack y sus dependencias

Fuente: (LoopBack , 2017)

LoopBack se compone de un grupo de módulos que pueden ser empleados de manera independiente o en conjunto para crear APIs REST de forma rápida. La Tabla 9 muestra las categorías de los módulos de LoopBack, su descripción y utilidad. Cada categoría incluye varios módulos, por ejemplo Integration contiene a loopback-connector-rest, loopback-connector-soap.

Tabla 9. Principales módulos de LoopBack

Categoría	Descripción	Utilidad
Models (Modelos)	Modelo y servidor API	Diseño dinámico de modelos, exhibirlos como API sin preocuparse por la persistencia.
Abstraction (Abstracción)	Modelo de abstracción de datos a persistencia física	Conexión a múltiples fuentes de datos o servicios y obtener un modelo abstracto con capacidades CRUD independientes de la fuente de datos de respaldo.
Initialization (Inicialización)	Inicialización de la aplicación	Configurar fuentes de datos, personalizar modelos, configurar modelos y conectarlos a fuentes de datos; configurar la aplicación y ejecutar scripts de arranque personalizados.
Sequencing (Secuenciación)	Ejecución de Middleware	Configuración de middleware para ser ejecutado en varios puntos durante el ciclo de vida de la aplicación.
Data (Datos)	Fuentes de datos físicos RDBMS y noSQL	Habilitar conexiones a fuentes de datos RDBMS y no SQL y obtener un modelo abstraído.
Integration (Integración)	Conectores de sistema general	Conexión a un sistema existente que expone APIs a través de las interfaces empresariales y web comunes
Components (Componentes)	Agregar complementos al núcleo LoopBack	Integrar con servicios pre-construidos empaquetados en componentes.
Clients (Clientes)	Kit de desarrollo de software (SDK) de cliente	Desarrollar la aplicación del cliente utilizando objetos nativos de la plataforma (iOS, Android, AngularJS) que interactúen con las APIs de LoopBack a través de REST.

Fuente: (LoopBack , 2017)

LoopBack incluye un cliente integrado llamado API Explorer que permite visualizar y probar los resultados de las APIs creadas en las aplicaciones LoopBack; es una excelente herramienta en tiempo de desarrollo, sin embargo es recomendable su desactivación en producción. La URL de acceso a LoopBack API Explorer de manera local es <http://localhost:3000/explorer/>.

CAPÍTULO III

3. MARCO METODOLÓGICO

Este capítulo presenta el resultado del análisis de información referente a la gestión de contenidos en el portal web de la UTN, para ello se efectuaron entrevistas y se realizó la revisión de la estructura de los blogs de carrera. En las primeras secciones se describe el lugar de desarrollo de la investigación, nivel y tipo, métodos, técnicas e instrumentos de recolección de datos; todos ellos determinados en concordancia con el objetivo planteado.

Las reuniones iniciales mantenidas con el Director de Desarrollo Tecnológico e Informático y el Web máster de la UTN permitieron establecer los instrumentos de recolección de datos e identificar a los actores involucrados; los instrumentos diseñados incluyeron cuestionarios de encuesta, entrevistas y formatos para la revisión de contenidos de los blogs, mediante ellos se estableció qué tipo de información se encuentra publicada y es potencial candidata para ser extraída del SIIU con el fin de alimentar el blog de la carrera de Ingeniería de Sistemas Computacionales.

El análisis realizado permitió identificar la información que puede ser compartida entre el SIIU y blog de la CISIC, que constituye uno de los insumos principales para la etapa de implementación de la propuesta presentada en el capítulo siguiente.

3.1. Lugar de desarrollo de la investigación

La investigación fue desarrollada en la Universidad Técnica del Norte (UTN) de la ciudad de Ibarra, provincia de Imbabura.

3.1.1. Unidad Ejecutora

La unidad ejecutora es la Dirección de Desarrollo Tecnológico e Informático (DDTI) de la UTN; su misión, visión y estructura organizacional se citan a continuación:

Misión.- “A la Dirección de Informática de la Universidad Técnica del Norte, le corresponde administrar los servicios centrales de informática, computación y comunicaciones, y en especial apoyar a las Facultades y otras Unidades de la Universidad, en temas del área; sin perjuicio de las demás funciones que se le encomiende” (Dirección de Informática, 2013).

Visión.- “La Dirección de Informática, en el año 2017 será quien ejerza el liderazgo a nivel institucional, regional y nacional en el campo de la informática, computación y telecomunicaciones con tecnología de punta, investigaciones de avanzada e innovación que aportará para la transformación de la UTN” (Dirección de Informática, 2013).

En el Plan de Desarrollo Tecnológico e Informático 2013-2017 se incluye el organigrama de la dependencia (Figura 18), en el mismo constan las áreas de Desarrollo de Sistemas de Información; Operación, Control y Soporte; Administración de Redes y Comunicaciones; Web y Desarrollo de nuevas Tecnologías de Información y Comunicación; y Centro Entrenamiento y Certificación Internacional.

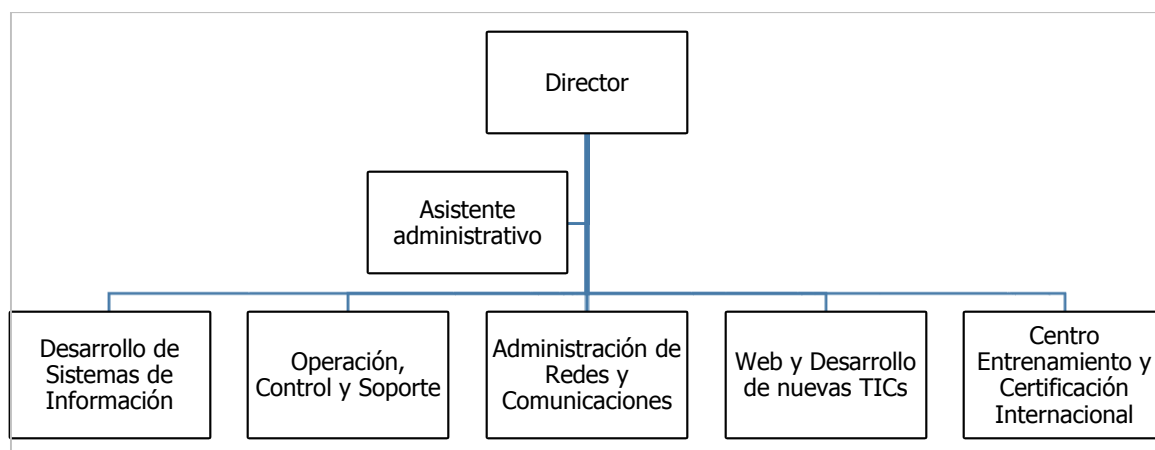


Figura 18. Organigrama del departamento de Desarrollo Tecnológico e Informático

Fuente: (Dirección de Informática, 2013)

La Dirección de Desarrollo Tecnológico e Informático cuenta con una estructura orgánica bien definida, en el Plan de Desarrollo Tecnológico e Informático 2013-2017 también se detallan las funciones establecidas para cada área.

3.1.2. Ubicación

- **Provincia:** Imbabura
- **Cantón:** Ibarra
- **Dirección:** Av. 17 de Julio 5-21 y Gral. José María Córdova
- **Denominación:** Universidad Técnica del Norte
- **E-mail:** info@utn.edu.ec
- **Teléfono:** +593 (6) 2 99 78 00

3.1.3. Beneficiarios

Beneficiarios directos: Estudiantes y docentes de la CISIC (319 y 23 respectivamente, según datos proporcionados por la Dirección de Desarrollo Tecnológico e Informático, 2016).

3.2. Modalidad de investigación

Este trabajo de investigación fue de tipo documental y de campo, la primera porque requirió la revisión de fuentes secundarias como libros y documentos digitales para el desarrollo del capítulo correspondiente a la fundamentación teórica; de campo porque se realizó en el lugar de los hechos y bajo una realidad definida.

3.3. Nivel o tipo de investigación

Según la finalidad, la investigación fue aplicada porque en base a los fundamentos científicos ya establecidos, se implementó una solución informática encaminada a resolver un problema práctico. De acuerdo a la profundidad la investigación fue descriptiva porque caracterizó el objeto de estudio, también se recolectó información que fue analizada estableciendo los resultados que permitieron emitir conclusiones.

3.4. Población y muestra

La población considerada para la investigación se detalla en la Tabla 10, constan el Web máster UTN, administradores de módulos del SIIU y el Director del Departamento de Desarrollo Tecnológico e Informático:

Tabla 10. Población

Descripción	Frecuencia
Web master UTN	1
Administradores de módulos Académico (ACA), Biblioteca (BIB), Planificación estratégica institucional (PEI), Oficina de Relaciones Internacionales (ORI), Recursos humanos (RHU), y Vinculación (VIN)	6
Director del Departamento de Desarrollo Tecnológico e Informático (DDTI)	1
Estudiantes CISIC	319
Docentes CISIC	23
Total	350

Fuente: Elaboración propia.

Se aplicó entrevistas a los administradores de los módulos identificados del SIIU, web master y Director del DDTI debido a su conocimiento de la información gestionada por las aplicaciones a ser integradas. Para el caso de docentes y estudiantes de la CISIC se utilizó una encuesta que requirió el cálculo de la muestra, empleando la fórmula siguiente:

$$n = \frac{N\sigma^2 Z^2}{(N-1)e^2 + \sigma^2 Z^2}$$

En donde:

n = el tamaño de la muestra.

N = tamaño de la población (docentes + estudiantes).

σ = Desviación estándar de la población (0,5).

Z = Valor obtenido mediante niveles de confianza (95% \rightarrow 1,96)

e = Límite aceptable de error muestral (0,05)

Utilizando la fórmula, la muestra es:

$$n = \frac{342*(0.5)^2(1.96)^2}{(342-1)(0.05)^2+(0.5)^2(1.96)^2} = 182$$

3.5. Operacionalización de variables

La Tabla 11 presenta la operacionalización de variables, que permitió identificar las fuentes de datos primarias y secundarias, así como las técnicas e instrumentos utilizados durante la recopilación de información.

Tabla 11. Operacionalización de variables

Variable independiente: Integración de Oracle y WordPress				
Conceptualización	Dimensiones	Indicadores	Ítems básicos	Técnicas e instrumentos
Proceso que permite vincular aplicaciones, en el nivel más sencillo comprende el intercambio de datos y/o funciones.	Nivel de integración entre el SIIU (Oracle) y el blog la CISIC (WordPress)	<ul style="list-style-type: none"> - Características de la integración de aplicaciones orientada a la información - Tecnologías que permiten integrar aplicaciones para compartir información - Módulos y tablas del SIIU que gestionan información para ser vinculada con el blog 	<ul style="list-style-type: none"> - ¿Cuáles son las características de la integración de aplicaciones orientada a la información? - ¿Qué métodos y tecnologías permiten integrar aplicaciones para compartir información? - ¿Cuál es la información del SIIU que puede emplearse para actualizar el blog de la CISIC del Portal Web? 	Revisión bibliográfica Entrevista Instrumento para la revisión de contenidos de los blogs de carreras
Variable dependiente: Actualización de contenidos en el portal web de la UTN				
Conceptualización	Dimensiones	Indicadores	Ítems básicos	Técnicas e instrumentos
La actualización de contenidos es parte de la gestión de un CMS; consiste en la modificación de la información publicada.	Categorías y contenidos del blog candidatos a ser actualizados mediante el SIIU	<ul style="list-style-type: none"> - Blogs de carrera del portal web - Descripción de categorías y contenidos del blog de la CISIC - Frecuencia de visita del blog y sus elementos - Estructura de la base de datos de WordPress - API Rest de WordPress 	<ul style="list-style-type: none"> - ¿Cuáles son los blogs de carrera del portal web de la UTN y qué contenidos se encuentran publicados? - ¿Cuáles son los contenidos publicados en el blog que pueden ser actualizados mediante el SIIU? - ¿Qué información del blog de la CISIC es visitada con mayor frecuencia? - ¿Qué tabla de WordPress almacena los posts? - ¿Cómo actualizar posts de WordPress a través de su API Rest? 	Revisión bibliográfica Instrumento para la revisión de contenidos de los blogs de carreras Encuesta

Fuente: Elaboración propia.

3.6. Métodos

La investigación propuesta requirió la aplicación de métodos que permitieron efectuar la búsqueda, selección y análisis de información proveniente de diferentes fuentes. El método analítico-sintético se empleó para identificar los elementos de la integración de aplicaciones posibilitando la compilación de la fundamentación teórica, también en la identificación de ítems pertenecientes a las variables independiente y dependiente. Una vez que fueron aplicadas las técnicas de recolección de datos se procedió a la interpretación de información, identificando los resultados más relevantes.

La inducción-deducción permitió el razonamiento, que en base a los resultados obtenidos mediante las técnicas de recolección de información y considerando las bases teóricas consultadas se establecieron las implicaciones generales en relación a las variables del tema de investigación.

3.7. Técnicas e instrumentos

Las técnicas de investigación que permitieron recolectar información proveniente de los actores relacionados con el tema, los fines alcanzados con cada una, así como los criterios empleados en su diseño se explican enseguida:

- **Encuesta.**- Aplicada a docentes y estudiantes de la CISIC para conocer su criterio en relación a los contenidos publicados en el blog de carrera (ver Anexo 2 – D).
- **Entrevista.**- Considerada una de las técnicas que propicia el diálogo con otras personas y permite obtener información de una fuente primaria. Esta técnica se aplicó al Director de Desarrollo Tecnológico e Informático, al Web máster institucional y a los administradores de los módulos de interés del SIIU (ver Anexo 2 - A, B y C respectivamente).
- **Observación.**- Se estableció un formato (ver Anexo 1) para identificar los contenidos de los blogs de carrera de la UTN, a fin de determinar qué tan común es cada contenido. En el caso específico del blog de la CISIC se diseñó un instrumento (Tabla 21) empleado para describir los contenidos informativos con sus respectivos formatos de publicación, identificando los módulos y tablas del SIIU de las que se podría obtener los datos de interés.

3.8. Recolección de la información

El proceso de recolección de información comprendió las siguientes actividades:

- **Sondeo preliminar.**- Se mantuvo reuniones de trabajo con el Director de Desarrollo Tecnológico e Informático así como con el web máster de la UTN para contextualizar el problema y caracterizar de manera general la vinculación entre el Sistema Informático Integrado Universitario y el portal web, lo que permitió identificar a los actores involucrados y establecer las técnicas de recolección de información a ser empleadas.
- **Aplicación de encuestas y entrevistas.**- Una vez operacionalizadas las variables se procedió a diseñar una primera versión de los cuestionarios, posteriormente se revisaron y ajustaron para su aplicación. Las entrevistas se destinaron a los administradores de los módulos identificados del SIIU, Director de DDTI y web máster de la universidad. Las encuestas fueron dirigidas a docentes y estudiantes de la CISIC.
- **Revisión de contenidos de blogs.**- Esta actividad comprendió la revisión del contenido de 36 blogs de carrera, identificando estructura, elementos comunes, tipos y formatos de información publicada; posteriormente se contrastó dichos datos con los obtenidos en las entrevistas para determinar los contenidos informativos a ser considerados como candidatos para vincularlos entre el SIIU y portal web.

Las etapas definidas para la recolección de información se desarrollaron de manera secuencial logrando así el cumplimiento de los objetivos trazados para cada una.

3.9. Análisis e interpretación de resultados

A continuación, se presenta los resultados obtenidos de las encuestas, entrevistas y revisión de blogs de carrera de la UTN, al finalizar la sección se incluyen las conclusiones y recomendaciones respectivas.

3.9.1. Resultados de la encuesta

Pregunta 1: Seleccione la frecuencia con la que accede al blog de la CISIC.

Tabla 12. Tabulación – Encuesta Pregunta 1

Opciones	Frecuencia	Porcentaje
Muy frecuentemente	25	14
Frecuentemente	68	37
Ocasionalmente	44	24
Raramente	22	12
Nunca	23	13
Total	182	100

Fuente: Elaboración propia.

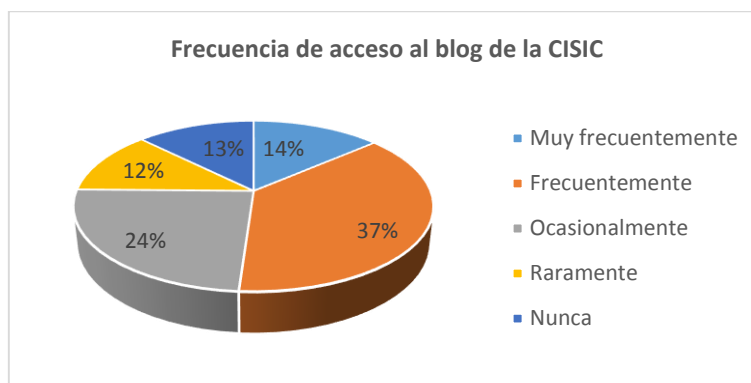


Figura 19. Encuesta – Pregunta 1

Fuente: Elaboración propia.

Análisis e interpretación.- La mayoría de encuestados indicaron que acceden al blog de carrera con mayor o menor frecuencia; sin embargo, un 13% afirmó nunca haber visitado el mismo, situación que podría mejorar mediante la socialización del blog en la CISIC.

Pregunta 2: ¿Considera que el blog de la CISIC cumple con el objetivo de informar sobre la carrera y sus actividades?

Tabla 13. Tabulación – Encuesta Pregunta 2

Opciones	Frecuencia	Porcentaje
Si	129	71
No	9	5
No sabe	44	24
Total	182	100

Fuente: Elaboración propia.

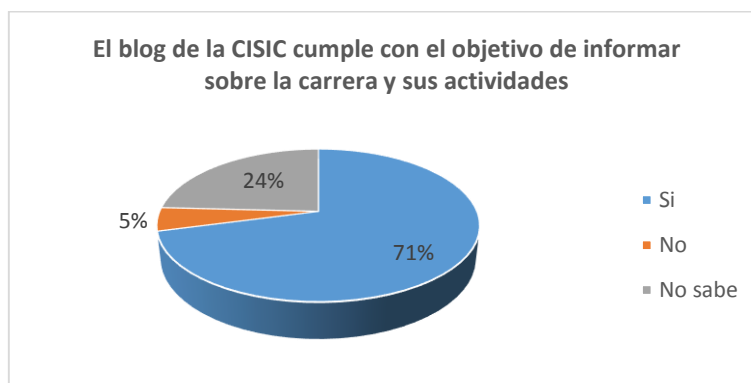


Figura 20. Encuesta – Pregunta 2
Fuente: Elaboración propia.

Análisis e interpretación.- La mayoría de encuestados (71%) coinciden en que el blog cumple con el objetivo de informar sobre la carrera y sus actividades; pero casi la cuarta parte de consultados desconoce si éste obedece a dicha tarea, por tanto surge la necesidad de reforzar la socialización del blog (sus objetivos y contenidos disponibles) hacia la comunidad universitaria de la CISIC.

Pregunta 3: ¿Cuál es el nivel de importancia que usted atribuye al blog para la Carrera?

Tabla 14. Tabulación – Encuesta Pregunta 3

Opciones	Frecuencia	Porcentaje
Muy importante	58	32
Importante	84	46
Medianamente importante	27	15
Poco importante	6	3
Nada importante	7	4
Total	182	100

Fuente: Elaboración propia.

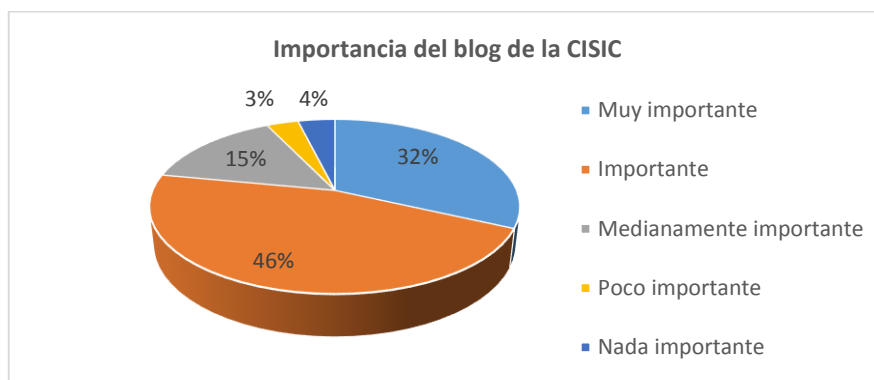


Figura 21. Encuesta – Pregunta 3
Fuente: Elaboración propia.

Análisis e interpretación.- El blog de carrera es considerado muy importante e importante para el 32% y 46% de encuestados respectivamente; un porcentaje bajo lo supone poco o nada importante. Los resultados permiten determinar que el blog si es conocido y se le atribuye importancia debido a que es un medio de difusión para la carrera.

Pregunta 4: ¿Considera que la información publicada en el blog de la CISIC se encuentra actualizada?

Tabla 15. Tabulación – Encuesta Pregunta 4

Opciones	Frecuencia	Porcentaje
Si	64	35
No	36	20
No sabe	82	45
Total	182	100

Fuente: Elaboración propia.

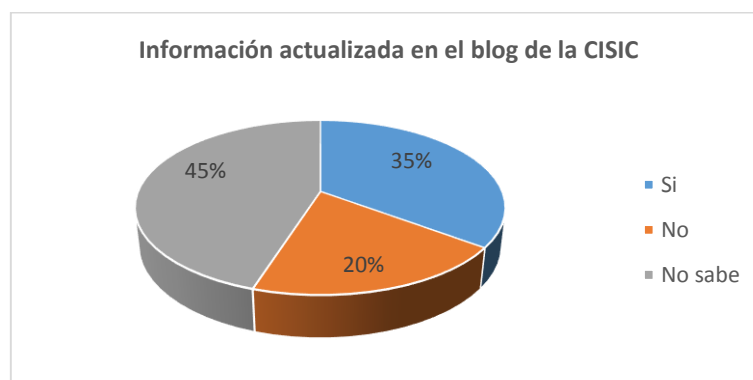


Figura 22. Encuesta – Pregunta 4
Fuente: Elaboración propia.

Análisis e interpretación.- El 20% de encuestados consideran que la información publicada en el blog no está actualizada, mientras que el 45% desconoce si lo está. Los resultados permiten concluir que, pese a que el blog es visitado, muchos de sus navegantes no conocen si la información es actual; además de la solución que permita actualizar los contenidos del blog desde el SIIU, se debería socializar que dicha actualización si se efectúa para que se genere mayor confianza en la información publicada.

Pregunta 5: De la categoría Información seleccione los ítems que usted consulta con mayor frecuencia en el blog de la CISIC.

Tabla 16. Tabulación – Encuesta Pregunta 5

Opciones	Frecuencia	Porcentaje
Misión/Visión de la carrera	110	60
Ficha de la carrera (Denominación, Coordinador, Correo electrónico)	73	40
Estructura organizacional	29	16
Plan de la carrera	71	39
Carrera en cifras	50	27

Fuente: Elaboración propia.

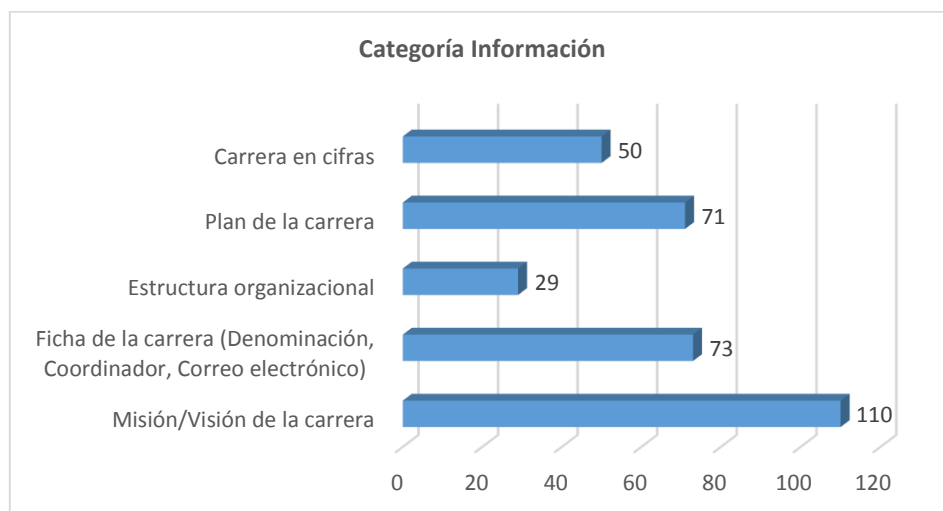


Figura 23. Encuesta – Pregunta 5

Fuente: Elaboración propia.

Análisis e interpretación.- En la categoría Información los elementos con mayor frecuencia de visita son Misión/Visión, ficha y plan de carrera; cuya actualización debería

llevarse a cabo en el caso de cambios de autoridades de la CISIC, o cuando se formule un nuevo plan estratégico de carrera que conlleve cambios en la información publicada. La estructura organizacional de la carrera tiene la frecuencia de visita más baja.

Pregunta 6: De la categoría GESTIÓN ACADÉMICA seleccione los ítems que usted consulta con mayor frecuencia en el blog de la CISIC.

Tabla 17. Tabulación – Encuesta Pregunta 6

Opciones	Frecuencia	Porcentaje
Perfil de ingreso/egreso/profesional	44	24
Objetivos educacionales	44	24
Resultados de aprendizaje	48	26
Malla curricular	121	66
Plan de estudios/Sílabos	85	47

Fuente: Elaboración propia.

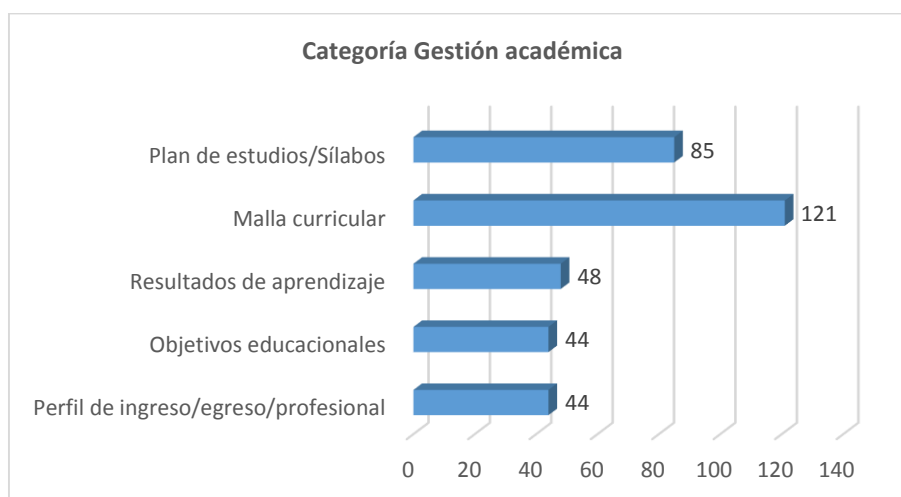


Figura 24. Encuesta – Pregunta 6

Fuente: Elaboración propia.

Análisis e interpretación. - Los contenidos de la categoría Gestión Académica del blog de la CISIC con mayor frecuencia de visita son Malla curricular y plan de estudios/sílabos; el primero corresponde a información propia de la carrera que debería ser actualizada al inicio de cada semestre, mientras que el segundo es un enlace a una aplicación externa al blog. Objetivos educacionales y los perfiles de ingreso/egreso/profesional son los contenidos menos visitados.

Pregunta 7: De la categoría INVESTIGACIÓN seleccione los ítems que usted consulta con mayor frecuencia en el blog de la CISIC.

Tabla 18. Tabulación – Encuesta Pregunta 7

Opciones	Frecuencia	Porcentaje
Planes/Políticas/Líneas/Programas/Redes de Investigación	37	20
Proyectos de Investigación	87	48
Docentes investigadores/Producción Científica	63	35
Trabajos de Titulación	83	46

Fuente: Elaboración propia.

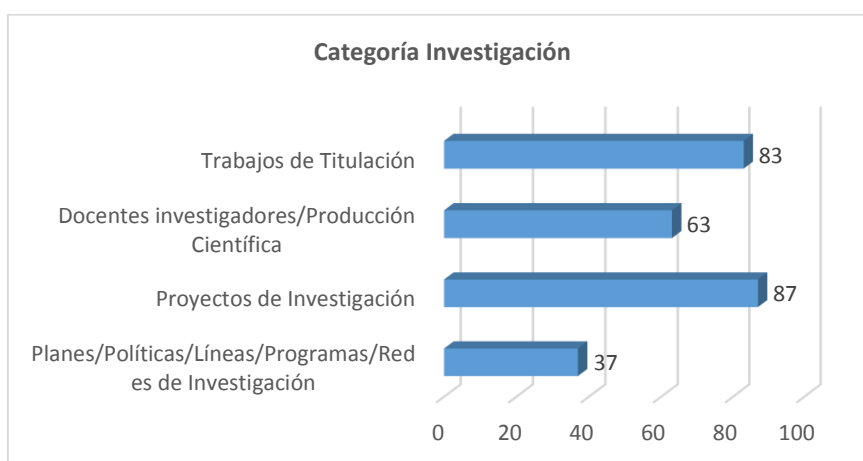


Figura 25. Encuesta – Pregunta 7

Fuente: Elaboración propia.

Análisis e interpretación.- Para el caso de la categoría Investigación, los elementos Proyectos de investigación, Trabajos de titulación y Docentes investigadores/Producción científica son los más frecuentemente visitados; dichos elementos deberían ser actualizados al iniciar un nuevo semestre académico, o cuando se generen nuevos proyectos de investigación, o finalicen/publiquen nuevos trabajos de grado, o en el caso de que se generen nuevas publicaciones por parte de los docentes de la CISIC. Planes/Políticas/Líneas/Programas/Redes de Investigación son los que se visitan con menor frecuencia.

Pregunta 8: De la categoría VINCULACIÓN seleccione los ítems que usted consulta con mayor frecuencia en el blog de la CISIC.

Tabla 19. Tabulación – Encuesta Pregunta 8

Opciones	Frecuencia	Porcentaje
Plan de vinculación	53	29
Docentes en la Vinculación	46	25
Movilidad Docente/Estudiantil	53	29
Convenios	94	52
Prácticas Pre-profesionales	87	48
Giras Académicas	54	30
Graduados CISIC	23	13

Fuente: Elaboración propia.

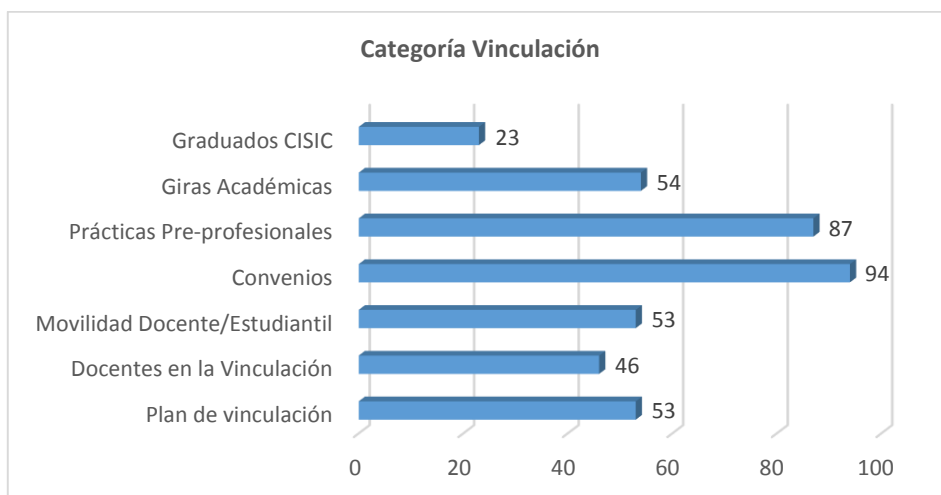


Figura 26. Encuesta – Pregunta 8

Fuente: Elaboración propia.

Análisis e interpretación.- Los ítems de la categoría Vinculación que son visitados con mayor frecuencia son convenios, prácticas pre-profesionales y giras académicas; estos deberían actualizarse en el blog conforme en la CISIC se firmen nuevos convenios, se modifiquen/generen informes de prácticas pre-profesionales y giras académicas de la carrera. El contenido Graduados CISIC es el que tiene una menor frecuencia de visitas con un porcentaje de 13%.

3.9.2. Resultados de las entrevistas

En esta sección se incluyen los resultados obtenidos de las entrevistas aplicadas al web máster, Director de DDTI y administradores de los módulos identificados del SIIU.

3.9.2.1. Entrevista aplicada al Web máster institucional

La entrevista enfocada al web máster tuvo el objetivo de obtener información acerca del proceso de gestión de contenidos en el portal web de la UTN, los datos recabados son:

- En el portal web se publica información institucional, noticias, anuncios, oferta académica de grado y posgrado, recursos y accesos a las aplicaciones informáticas de la universidad. Debido a la cantidad de carreras e información relativa a cada una, en el DDTI se optó por diseñar un micro-sitio o blog por cada carrera, actualmente se cuenta con 37 según se detalla en la Tabla 20:

Tabla 20. *Blogs de carreras por facultad*

Facultad	Carreras	Frecuencia	
Facultad de Ingeniería en Ciencias Aplicadas (FICA)	<ul style="list-style-type: none"> • Electrónica • Sistemas • Mecatrónica • Industrial 	<ul style="list-style-type: none"> • Textil • Automotriz • Eléctrica 	7
Facultad de Ciencias Agropecuarias y Ambientales (FICAYA)	<ul style="list-style-type: none"> • Agroindustrias • Agronegocios • Agropecuaria • Recursos Naturales 	<ul style="list-style-type: none"> • Forestal • Bio-tecnología • Energías Renovables 	7
Facultad de Ciencias Administrativas y Económicas (FACAE)	<ul style="list-style-type: none"> • Administración • Contabilidad • Economía • Mercadotecnia 	<ul style="list-style-type: none"> • Gastronomía • Turismo • Derecho 	7
Facultad de Ciencias de la Salud (FCSALUD)	<ul style="list-style-type: none"> • Enfermería • Terapia física 	<ul style="list-style-type: none"> • Nutrición 	3
Facultad de Educación, Ciencia y Tecnología (FECYT)	<ul style="list-style-type: none"> • Psicología educativa • Psicología general • Artes plásticas • Diseño y publicidad • Diseño gráfico • Gestión y desarrollo social • Relaciones públicas 	<ul style="list-style-type: none"> • Secretariado ejecutivo • Físico matemático • Inglés • Parvularia • Entrenamiento deportivo • Educación física 	13
Total		37	

Fuente: Elaboración propia.

- El contenido de los blogs se encuentra estandarizado en la mayoría de casos y está organizado en cuatro categorías. La categoría *Información* incluye ficha, historia, misión y visión, entre otros. La categoría *Gestión académica* cuenta con perfiles de ingreso/egreso/profesional, campo ocupacional, objetivos educacionales, resultados de aprendizaje y malla curricular. La categoría *Investigación* comprende el plan/políticas/líneas/programas/redes/proyectos de investigación, entre otros. La categoría Vinculación contiene plan de vinculación, movilidad docente y estudiantil, convenios, entre otros.
- En el Departamento de Desarrollo Tecnológico e Informático se ha designado a una persona como web máster; debido a la cantidad de blogs de carrera la gestión de contenidos informativos constituye una tarea demandante.
- Los blogs de carrera están implementados con la plataforma Wordpress 4.7, servidor web Apache 2.2, base de datos MySQL 5.6.13.

El web máster considera que el desarrollo de una aplicación encaminada a vincular el SIIU con el blog de la CISIC permitirá aprovechar la información del SIIU con el fin de actualizar los contenidos informativos en el blog; además de identificar el mecanismo de vinculación que puede ser replicado con el resto de blogs disponibles facilitando así la gestión de contenidos informativos en los blogs de carrera del portal web.

3.9.2.2. Entrevista realizada al Director del DDTI

La entrevista aplicada buscó la identificación de módulos del SIIU cuya información coincide con la publicada en el blog de carrera. La persona entrevistada se desempeñó anteriormente como Jefe de proyectos por tanto conoce a detalle el diseño del SIIU. La información obtenida se explica a continuación:

- El Sistema Informático Integrado Universitario es el ERP institucional, está conformado por módulos que apoyan las tareas universitarias en los ámbitos de gestión, docencia, investigación y vinculación con la colectividad. Es un desarrollo in-house que comenzó en el año 2009 y al que paulatinamente se han ido incorporando módulos que cubren requerimientos de diferentes dependencias universitarias.

- El SIIU cuenta con 27 módulos de los cuales: Gestión académica (ACA), Recursos humanos (RHU), Planeamiento estratégico institucional (PEI), Gestión de biblioteca (BIB), Oficina de Relaciones Internacionales (ORI) y Vinculación con la colectividad (VIN) gestionan información que podría servir como insumo para actualizar los contenidos informativos de los blogs de carrera; de los módulos listados, el académico es el que mayor cantidad de información puede aportar en la tarea mencionada. Datos como misión y visión, malla curricular, resultados de aprendizaje, autoridades, anteproyectos aprobados, graduados y egresados, y estadísticas de indicadores de gestión puede ser tomados del SIIU para ser actualizados en el blog de carrera.
- El SIIU está implementado con la plataforma Oracle (Oracle Forms 11g, Oracle Reports 11g, Oracle Apex 4.2, Servidor de Aplicaciones Web Logic 11, Base de datos Oracle 11g).

El Director del DDTI considera que la implementación de una aplicación encaminada a vincular el SIIU con el blog de la CISIC permitirá que el blog esté actualizado y a la par con la información almacenada en el SIIU. Una vez que se haya establecido la manera de vincular la información del SIIU con el blog de la CISIC, el DDTI podría replicar el mismo proceso para otros blogs.

3.9.2.3. Entrevista a los administradores de los módulos del SIIU

La entrevista enfocada a los administradores de los módulos ACA, BIB, ORI, RHU, PEI y VIN se realizó con el objetivo de identificar cuáles de los datos publicados en el blog de la CISIC coinciden con la información almacenada en la base de datos del SIIU y por tanto podrían compartirse entre las aplicaciones, se obtuvo los siguientes datos:

- El módulo académico gestiona objetivos educacionales, resultados de aprendizaje, malla curricular. En conjunto con el módulo de recursos humanos gestiona distributivo docente, obras publicadas, listado de graduados, ficha de carrera (nombre del coordinador y correo electrónico, denominación de carrera, modalidad, sistema de estudios, tipo de ciclo académico y financiamiento).

- La información sobre los trabajos de titulación está disponible en el módulo de biblioteca; del módulo de planificación estratégica institucional es posible obtener la misión y visión de un determinado centro de costo de la universidad. El módulo de ORI gestiona el listado de convenios institucionales.
- De entre los módulos revisados el que mayor cantidad de información puede aportar es el módulo académico. Los scripts SQL que permiten obtener la información de interés del SIU pueden ser parametrizados por dependencia, de manera que se obtiene la información de una determinada carrera.

Las entrevistas permitieron identificar los blogs que forman parte del portal web institucional, su estructura y contenidos; también la información gestionada por los módulos del SIIU que podría ser candidata para ser compartida con los blogs de carrera. Los entrevistados coinciden en que sería beneficioso la implementación de una aplicación para compartir datos entre el SIIU y blog de la CISIC debido a que en el DDTI se podría repetir el mismo proceso con el resto de blogs, se facilitaría la publicación de contenidos además de que la información del SIIU y blogs estaría a la par.

3.9.3. Revisión de contenidos de los blogs

La oferta académica publicada en el portal web de la UTN incluye links a 37 blogs de carrera que se encuentran agrupados por facultad. La revisión excluyó a la Carrera de Inglés debido a que sus contenidos están publicados en ese idioma. Los contenidos de los blogs están organizados en cuatro categorías: Información, Gestión Académica, Investigación y Vinculación; cada una contiene varios elementos según se indican en el Anexo 1.

La Figura 27 muestra que el 83% de blogs revisados sigue la estructura con las 4 categorías anteriormente descritas, 6% no se ajustan a ella y el 11% no estuvieron disponibles las ocasiones que se realizó la revisión; estos datos hacen posible concluir que la mayoría de blogs de carrera de la UTN sigue una estructura estándar.

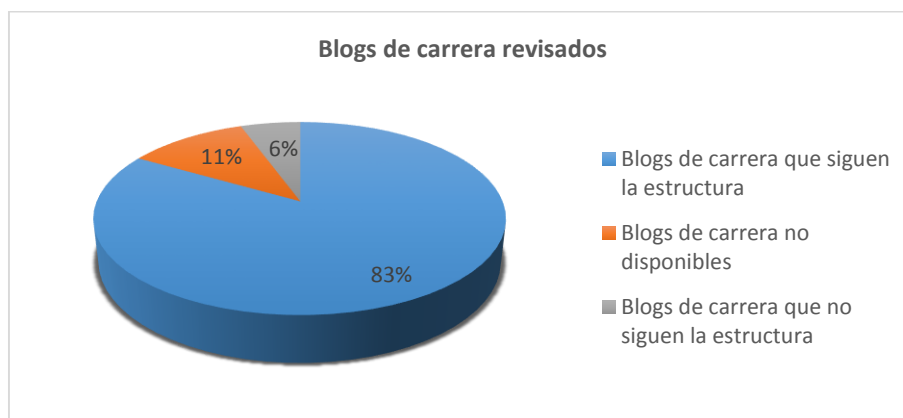


Figura 27: Revisión de blogs de carrera

Fuente: Instrumento de revisión de blogs de carreras (Anexo 1)

En el Anexo 1 se identificaron 50 ítems, el 76% de ellos corresponde a contenidos informativos es decir datos propios de la carrera en diferentes ámbitos, mientras que el 24% a enlaces hacia sitios web externos u otras aplicaciones (Figura 28).

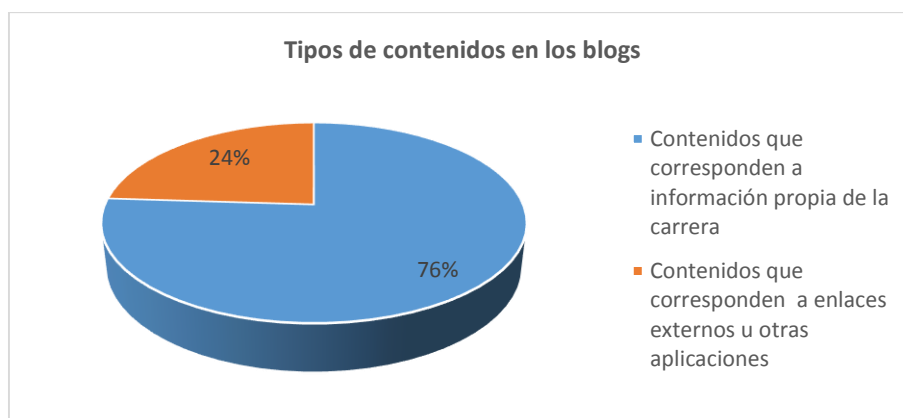


Figura 28: Tipos de contenidos en los blogs de carrera

Fuente: Instrumento de revisión de blogs de carreras (Anexo 1)

La Figura 29 presenta el porcentaje de ocurrencia de los contenidos en los blogs de carrera, los contenidos informativos de mayor ocurrencia son A3.- Historia de la Carrera, A4.- Misión y visión, A5.- Estructura Organizacional, B1.- Perfil de Ingreso, B5.- Campo Ocupacional y B7.- Resultados de Aprendizaje con 83%. Mientras que los ítems menos comunes son A8.- Docentes de la Carrera, A10.- Datos Generales, B9.- Comisión Asesora, B10.- Formato CD, B14.- Blog Docente, Web Services e Indicadores de Carrera con 3%.

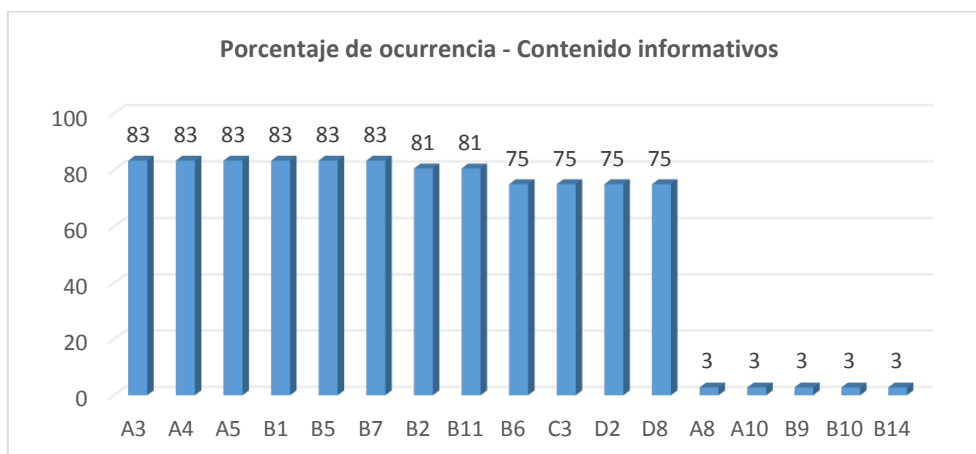


Figura 29: Porcentaje de ocurrencia de los contenidos

Fuente: Instrumento de revisión de blogs de carreras (Anexo 1)

La Tabla 21 presenta el resultado de la revisión pormenorizada del blog de la CISIC junto a los datos obtenidos en las entrevistas, los campos incluidos son:

- **Enlaces.**- Es el nombre del contenido informativo del blog, tal y como aparece en el menú de opciones.
- **Descripción de contenidos.**- Especifica la naturaleza de la información publicada.
- **Tipo de contenido publicado.**- Señala el formato de publicación (HTML con texto e imágenes, documentos PDF o formatos como DOC y XLS).
- **Disponible en el SIIU.**- Indica la disponibilidad del contenido en el SIIU con el valor *Si* para cuando está disponible, *No* cuando no lo está, *parc.* para cuando está parcialmente disponible y *n/a* cuando no aplica para el caso de enlaces a sitios o aplicaciones externos.
- **Candidato a ser extraído.**- Especifica si la información disponible en el SIIU será compartida desde él hacia el blog.
- **Módulo fuente y tablas.**- Para los contenidos presentes en el SIIU se indica las siglas de los módulos que los gestionan, y las tablas que contienen la información en cuestión.

Tabla 21. Revisión de contenidos del blog de la CISIC

Enlaces	Descripción de Contenidos	Tipo de contenido publicado				Disponible en el SIU	Candidato a ser extraído	Módulo Fuente	Tabla(s)	Observación
		Txt	Img	Pdf	App					
Nuestra Carrera	Fotografía coordinador, mensaje y hoja de vida	✓	✓	✓	x	n/d				
Ficha de Carrera	Denominación, modalidad, duración, resolución de aprobación, título que otorga, coordinador de carrera, email coordinador, email carrera, teléfono	✓	x	x	x	parc.	Si	RHU ACA	RHU_TAB_AUTORIDADES ACA_TAB_DEPENDENCIAS ACA_TAB_ESQUEMAS_ESTUDIOS_CICL ACA_TAB_MODALIDADES_ESTUDIOS ACA_TAB_SISTEMAS_ESTUDIOS ACA_TAB_TIPOS_CICLOS_ACADEMICO ACA_TAB_TIPOS_FINANCIAMIENTOS RHU_TAB_CORREO_INSTITUCIONAL	Información disponible Nombre_Coordinador Correo_Coordinador Denominación_Carrera Modalidad Sistema_Estudios Tipo_CiclosAcad Tipo_Financiamiento
Historia de la Carrera	Fotografía, historia, proyecto de creación de la carrera	✓	✓	✓	x	n/d				
Misión y visión	Misión y visión de la carrera	✓	x	x	x	Si	Si	PEI	PEI_TAB_MISION PEI_TAB_VISION	
Estructura Organizacional	Organigrama de la carrera	x	✓	✓	x	n/d				
Plan estratégico de carrera	Plan estratégico de la carrera (Datos informativos, misión, visión, políticas, mallas curriculares, distributivo, calendario, perfil de egreso, resultados y logros, líneas de investigación, proyectos de investigación, capacitaciones, programas y proyectos de vinculación)	x	x	✓	x	n/d				
Laboratorios/Infraestructura	Descripción laboratorios de informática de la facultad, cantidad de computadoras, organigrama, responsables, horarios de atención, funciones, áreas físicas, software disponible, fotografías	x	x	✓	x	n/d				

INFORMACIÓN

	Enlaces	Descripción de Contenidos	Tipo de contenido publicado				Disponible en el SIU	Candidato a ser extraído	Módulo Fuente	Tabla(s)	Observación
			Txt	Img	Pdf	App					
	Carrera en Cifras	Enlace a aplicación con APEX	x	x	x	✓	n/a				
GESTIÓN ACADÉMICA	Perfil de Ingreso	Perfil de ingreso	✓	✓	✓	x	n/d				
	Perfil de Egreso	Perfil de egreso	✓	✓	x	x	n/d				
	Perfil Profesional	Perfil profesional	✓	✓	x	x	n/d				
	Redes Académicas	Redes académicas	✓	✓	x	x	n/d				
	Campo Ocupacional	Campo ocupacional	✓	✓	x	x	n/d				
	Objetivos Educativos	Objetivos educacionales	✓	✓	x	x	Si	Si	ACA	ACA_TAB_OBJ_EDUCACIONALES	
	Resultados de Aprendizaje	Resultados de aprendizaje	✓	✓	x	x	Si	Si	ACA	ACA_TAB_RESULT_APRENDIZAJE	
	Diseño/Rediseño Curricular	Documento de diseño y rediseño curricular	x	x	✓	x	n/d				
	Comisión Asesora	Integrantes comisión asesora	✓	✓	x	x	n/d				
	Formato CD	Modelo de CD, formato docx del artículo	✓	✓	✓	x	n/a				
	Malla Curricular	Malla curricular de la carrera (pdf - gráfico)	x	x	✓	x	Si	Si	ACA	ACA_TAB_MATERIAS_PENSUMS ACA_TAB_MATERIAS ACA_TAB_MATERIAS_PREREQUISITOS	
	Plan de Estudios	Enlace a aplicación APEX	x	x	x	✓	n/a				
	Sílabos por Materia	Enlace a aplicación APEX	x	x	x	✓	n/a				

	Enlaces	Descripción de Contenidos	Tipo de contenido publicado				Disponible en el SIIU	Candidato a ser extraído	Módulo Fuente	Tabla(s)	Observación
			Txt	Img	Pdf	App					
INVESTIGACIÓN	Plan de Investigación	Objetivos, líneas, sub-líneas	x	x	✓	x	n/d				
	Políticas de Investigación	Políticas de investigación de la carrera	✓	x	x	x	n/d				
	Líneas de Investigación	Líneas de investigación de la carrera	✓	x	x	x	n/d				
	Programas de Investigación	Programas de investigación de la carrera	x	x	✓	x	n/d				
	Redes de Investigación	Enlace externo a CEDIA	x	x	x	x	n/a				
	Proyectos de Investigación	Proyectos de investigación: Nombre del Proyecto, línea de investigación, carreras involucradas, fecha inicio, fecha fin, estado, alcance territorial, director del proyecto, equipo de trabajo (XLS)	x	x	x	x	n/d				
	Docentes investigadores/ Producción Científica	Nombres de docentes con carga horaria en investigación	✓	✓	x	x	Si	Si	ACA RHU	Carga horaria en investigación ACA_TAB_MATERIAS_PENSUMS ACA_TAB_DISTRIBUTIVOS BL_VIEW_NOMINA_DOCENTE RHU_TAB_PERSONAS Produccion científica ACA_TAB_DISTRIBUTIVOS RHU_TAB_OBRAS_PUBLICADAS RHU_TAB_PERSONAS	La consulta presenta la información de la carga horaria de los docentes de la carrera considerando horas de docencia, investigación, gestión y vinculación (Ítem D2)
	Investigación Formativa	No hay información publicada	x	x	x	x	n/a				
	Trabajos de Titulación	Acceso a repositorio institucional	x	x	x	x	Si	No	ACA BIB RHU	ACA_TAB_DEPENDENCIAS BIB_TAB_DOCUMENTOS BIB_TAB_AUTORES_DOCUMENTOS RHU_TAB_PERSONAS	La información se encuentra disponible en el SIIU pero no será compartida ya que a través del blog se accesa al repositorio institucional en donde está publicado el documento completo de cada trabajo de titulación

	Enlaces	Descripción de Contenidos	Tipo de contenido publicado				Disponible en el SIU	Candidato a ser extraído	Módulo Fuente	Tabla(s)	Observación
			Txt	Img	Pdf	App					
VINCULACIÓN	Plan de vinculación	Documento PDF que contiene el plan de investigación de la carrera	x	x	✓	x	n/d				
	Docentes en la Vinculación	Fotografía, texto datos en tabla con número de horas	✓	✓	X	x	Si	Si	ACA RHU	ACA_TAB_MATERIAS_PENSUMS ACA_TAB_DISTRIBUTIVOS BI_VIEW_NOMINA_DOCENTE RHU_TAB_PERSONAS	
	Cultura	Enlace a http://www.utn.edu.ec/cultura/	x	x	X	x	n/a				
	Club de programación CP - UTN	Fotografía y texto informativo sobre el club de programación de la CISIC	✓	✓	X	x	n/d				
	Movilidad Docente	Fotografía y texto	✓	✓	X	x	parc.	No	ORI	ORI_MOVILIDAD ORI_MOVILIDAD_PARTICIPANTES ORI_CONVENIO	Existe información sobre movilidad no existe un campo que indique si es convenio a nivel de docentes o estudiantes
	Movilidad Estudiantil	Fotografía y texto	✓	✓	X	x	parc.	No			
	Convenios	Texto	✓	x	X	x	Si	Si	ORI VIN	ORI_CONVENIO ORI_TIPO_CONVENIO ORI_CONVENIO_INSTITUCION ORI_CONVENIO_DEPENDENCIAS VIN_TAB_INSTITUCION	El listado obtenido del SIU contiene mayor grado de descripción del inicialmente publicado en el blog, por lo que se añadirá el contenido al ya existente en el blog
	Prácticas Pre-profesionales	Indicaciones en texto, formatos docx y pdf para descargar	✓	x	✓	x	n/d				
	Giras Académicas	Texto, fotografías, informes en formato pdf	✓	✓	x	x	n/d				
	Graduados CISIC	Listado de graduados por en los últimos 3 años	x	x	✓	x	Si	No	ACA RHU	ACA_TAB_ACTIVIDADES_ESTUDIANTE RHU_TAB_PERSONAS	El contenido corresponde a un extenso listado, cuya presentación en HTML no es idónea

Fuente: Elaboración propia.

No toda la información del blog de la CISIC y que está disponible en el SIIU será compartida entre las dos aplicaciones, los detalles en cada caso se describen en la columna *Observación*. La decisión de vincular o no la información entre el SIIU y el blog se basó en factores como el formato de publicación de los contenidos en el blog y la cantidad de información a desplegar.

3.9.4. Conclusiones y recomendaciones

Conclusiones

- Los contenidos informativos del blog más visitados son misión, visión y ficha de carrera en la categoría Información; malla curricular y plan de estudios/sílabos en la categoría Gestión académica; proyectos de investigación, trabajos de titulación y docentes investigadores/producción científica en la categoría Investigación; y convenios, prácticas pre-profesionales y giras académicas en la categoría Vinculación.
- El Sistema Integrado Universitario (SIIU) y el portal web son aplicaciones informáticas que cumplen funciones específicas y están implementadas bajo plataformas tecnológicas heterogéneas, dicha característica no ha posibilitado el aprovechamiento de información existente en el SIIU para actualizar publicaciones en los blogs de carreras.
- La información que se encuentra disponible en el SIIU y puede ser utilizada en el blog de la CISIC es gestionada por los módulos Académico, Biblioteca, Planificación estratégica institucional, Oficina de Relaciones Internacionales, Recursos humanos y Vinculación; y es la ficha de carrera (nombre del coordinador, correo electrónico del coordinador, denominación de la carrera, modalidad, sistema de estudios, tipo de ciclo académico, tipo de financiamiento), misión y visión, objetivos educacionales, resultados de aprendizaje, malla curricular, docentes investigadores/producción científica, trabajos de titulación, docentes en la vinculación, convenios y listado de graduados.
- La integración entre el SIIU y el portal web debe enfocarse en la automatización de tareas para actualizar contenidos informativos del blog de la CISIC, ahorrando

tiempo y facilitando su realización; de manera que la información albergada en el SIIU y la publicada en el blog estén a la par.

- El 83% de los blogs de carreras de la UTN sigue una estructura similar en lo referente a la organización de contenidos, y la información proveniente del SIIU puede ser extraída considerando como parámetro la dependencia (carrera) por tanto el DDTI podría replicar el ejercicio de vinculación entre el SIIU y el blog de la CISIC a otros blogs de carrera.

Recomendaciones

- Pese a que el blog de la CISIC es visitado con frecuencia y es considerado importante para la difusión de información, es necesario socializarlo junto a sus objetivos, contenidos publicados y periodicidad con la que se actualiza la información publicada.
- Se recomienda identificar el mecanismo que permite vincular el SIIU y blog de la CISIC, con la finalidad de compartir información entre ambas.
- Se recomienda seleccionar los elementos de cada categoría del blog de la CISIC que pueden ser actualizados desde el SIIU, identificando las tablas involucradas en cada aplicación además de los scripts SQL con los respectivos parámetros que permiten obtener la información del SIIU.
- Se recomienda la implementación de una aplicación encaminada a la integración de datos entre el SIIU y portal web, de manera que se aproveche en el portal la información disponible en el SIIU; facilitando así la actualización de contenidos previamente establecidos en el blog de la carrera de Ingeniería en Sistemas computacionales.

CAPÍTULO IV

4. PROPUESTA

La propuesta presentada en este capítulo incluye antecedentes, metodología de desarrollo, objetivos, arquitectura de integración, esquema del software, y el detalle de ejecución de los sprints. La integración del SIIU y blog de la CISIC buscó compartir información entre ambas aplicaciones, facilitando las tareas de actualización de contenidos en el blog aprovechando la información disponible en el SIIU. El software fue desarrollado de manera incremental e iterativa según el marco de trabajo Scrum para ello se definieron sprints orientados a la implementación de las funcionalidades requeridas. Cada sprint comprendió tres fases: planificación, ejecución y revisión. Al final del capítulo se pasa revista a los resultados alcanzados en cada sprint y se explica los beneficios logrados.

4.1. Antecedentes de la propuesta

En el blog de la Carrera de Ingeniería en Sistemas Computacionales los contenidos se encuentran agrupados en las categorías: Información, Gestión Académica, Investigación y Vinculación; cada una agrupa varios elementos según se detalla en la Figura 30:

Información ▼	Gestión Académica ▼	Investigación ▼	Vinculación ▼
Nuestra Carrera	Perfil de Ingreso	Plan de Investigación	Plan de vinculación
Ficha de Carrera	Perfil de Egreso	Políticas de Investigación	Docentes en la Vinculación
Historia de la Carrera	Perfil Profesional	Líneas de Investigación	Cultura
Misión y visión	Redes Académicas	Programas de Investigación	Club de programación CP - UTN
Estructura Organizacional	Campo Ocupacional	Redes de Investigación	Movilidad Docente
Plan estratégico de carrera	Objetivos Educativos	Proyectos de Investigación	Movilidad Estudiantil
Laboratorios/Infraestructura	Resultados de Aprendizaje	Docentes investigadores/Producción Científica	Convenios
Carrera en Cifras	Diseño/Rediseño Curricular	Investigación Formativa	Prácticas Pre-profesionales
	Comisión Asesora	Trabajos de Titulación	Giras Académicas
	Formato CD		Graduados CISIC
	Malla Curricular		
	Plan de Estudios		
	Sílabos por Materia		

Figura 30. Menús del blog de la CISIC

Fuente: Elaboración propia.

Varios de los contenidos publicados en el blog de carrera también lo están en el SIIU pero cada aplicación los gestiona de forma independiente; cuando la información “común” es modificada en el SIIU debería realizarse una tarea similar en el blog para actualizarla de manera que sea coincidente entre ambas; esta situación no siempre ocurre enseguida, ocasionando que los contenidos publicados en el blog no se actualicen oportunamente.

Una solución al problema mencionado consiste en la actualización manual del blog de la CISIC con información proveniente del SIIU, dicha tarea implicaría:

- Identificar los contenidos del blog que podrían ser actualizados con datos provenientes del SIIU; los módulos del SIIU que gestionan dicha información, además de las tablas/vistas y otros objetos de base de datos involucrados.
- Los administradores de los módulos del SIIU deberían alertar al web máster cuando se haga alguna modificación en la información identificada, obtenerla a partir de scripts SQL o tareas de exportación para después proveerla, de tal manera que el web máster esté pendiente de la actualización que debe ser realizada en el blog y cuente con la información para ello.
- El web máster podría importar los datos provistos, sin embargo este paso no resulta sencillo porque en el blog la información es almacenada junto a su formato de presentación en HTML. Otra alternativa a este último paso sería la actualización del blog desde la herramienta de gestión de contenidos del CMS.

El proceso manual descrito es complejo, lo expuesto ratificó la necesidad de implementar un software encaminado a compartir datos entre el SIIU y blog de carrera que apoye las tareas de actualización de contenidos en el blog de la CISIC, aprovechando los datos disponibles en el SIIU; logrando así que la información en ambos no se encuentre desfasada.

Mediante el diagnóstico realizado en el capítulo precedente se identificó la información común entre el SIIU y blog de la CISIC, los módulos del SIIU que la gestionan, además de los scripts de consulta con sus respectivos parámetros según la dependencia de interés; constituyéndose en los insumos para la propuesta que a continuación se describe.

4.2. Metodología de desarrollo de software

La metodología seguida para el desarrollo del software fue Scrum con ciertas modificaciones. Scrum es un marco de trabajo que sigue el manifiesto ágil, caracterizado porque el producto se divide en entregables parciales definidos en concordancia con las necesidades del cliente; en donde el equipo de desarrollo es auto-organizado, sigue un conjunto de tareas planificadas y evalúa su trabajo en busca de mejoras, adaptándose a requerimientos cambiantes o que inicialmente no están lo suficientemente claros.

Enseguida se describen las características de Scrum considerando los criterios de Schwaber y Sutherland (2013) y de Menzinsky, López y Palacio (2016), posteriormente se explica cómo se aplicaron en este proyecto:

- **Roles.-** En Scrum los tres principales roles son: *dueño del producto* (Product Owner) quien es la voz del cliente o los interesados, define los objetivos y tiene la visión del producto, participa en la planificación, determina la pila del producto. *Scrum master* apoya al equipo de desarrollo, es un facilitador eliminando obstáculos que puedan impedir la consecución de los objetivos trazados, asegura que el proceso Scrum se cumpla. *Equipo de desarrollo* (Development Team) encargado de implementar el producto, es auto-organizado y multifuncional.
- **Eventos.-** Los eventos definidos en Scrum son el sprint, reuniones de planificación de sprint, scrum diario, revisión de sprint y retrospectiva de sprint. Cada iteración en Scrum se conoce como *sprint*, usualmente dura de 2 a 4 semanas y tiene como resultado un incremento entregable del producto. En las *reuniones de planificación* (Sprint Planning Meeting) participan el dueño del producto, equipo de desarrollo y Scrum master, para definir objetivos a alcanzar (Sprint Goal) y tareas a desarrollar durante el sprint. El *Scrum diario* es una corta reunión de 15 minutos efectuada para que cada miembro del equipo de desarrollo exponga brevemente el trabajo realizado el día anterior, el que hará el día actual, además de sus necesidades o problemas; durante esta reunión la pila de sprint y su avance deben actualizarse. La *revisión de sprint* (Sprint Review) se realiza cuando éste finaliza, en ella participan el equipo de desarrollo, scrum master, dueño del

producto y otros invitados implicados en el proyecto; se verifica el incremento alcanzado y revisa la pila del producto. La *retrospectiva de sprint* (Sprint Retrospective) se efectúa después de la revisión del sprint y antes de la reunión de planificación de uno nuevo, tiene el objetivo de revisar cómo se está desempeñando el equipo identificando fortalezas y problemas con el fin de establecer un plan de mejoras.

- **Artefactos.**- Los artefactos de Scrum son pila de producto (Product Backlog), pila de sprint (Sprint Backlog) e incremento; la *pila del producto* es el conjunto de requerimientos priorizados del producto según su valor para el negocio, es modificado únicamente por el dueño del producto, crece y evoluciona conforme avanza el desarrollo. La *pila de sprint* describe las tareas necesarias para llevar a efecto lo planificado, es decir es el cómo se va a llevar a cabo el sprint. El *incremento* constituye las funcionalidades operativas desarrolladas en un sprint.

Una vez que se ha explicado de manera breve el marco de trabajo Scrum, se procede a indicar cómo fue empleado:

- El software fue implementado de manera iterativa e incremental. Los sprints comprendieron tres momentos: planificación, ejecución y revisión. Durante la planificación se identificaron las tareas y número de horas de trabajo para cumplir los objetivos; la ejecución comprendió el desarrollo de cada tarea conforme lo planificado realizando los ajustes de caso en cuanto al tiempo y fechas de inicio/finalización; la revisión del sprint permitió examinar el cumplimiento del incremento y establecer consideraciones que sirvieron como referente para planificar los siguientes sprints.
- En cuanto a los roles cabe señalar que no se efectuó cada uno tal como lo especifica Scrum, más bien las funciones adscritas a ellos fueron desempeñadas indistintamente por la tesista por tratarse de un trabajo de titulación individual.
- En lo que respecta a los artefactos de Scrum, se definió la pila del producto inicial que fue actualizándose al finalizar cada sprint; también se diseñaron las pilas de sprint con sus respectivas tareas, duración y estado.

4.3. Integración del SIIU y el blog de la CISIC (Solución propuesta)

El SIIU y portal web son aplicaciones heterogéneas, el primero está implementado con la plataforma Oracle 11g y conformado por módulos que gestionan la información académica, de gestión, de investigación y de vinculación con la colectividad. En el portal web además de noticias, acceso a aplicaciones, recursos y servicios, se publica información sobre las carreras que oferta la universidad mediante blogs desarrollados con WordPress y MySQL.

El objetivo principal de la integración entre el SIIU y blog de carrera es compartir datos entre ellos, dicha tarea debería efectuarse cuando los datos publicados en el blog no correspondan con los almacenados en la base de datos del SIIU. La información que está disponible en el SIIU y podría utilizarse para actualizar la publicada en el blog de la CISIC es gestionada por los módulos Académico, Biblioteca, Planificación estratégica institucional, Oficina de Relaciones Internacionales, Recursos humanos y Vinculación; y consiste en la ficha de carrera (nombre del coordinador, correo electrónico del coordinador, denominación de la carrera, modalidad, sistema de estudios, tipo de ciclo académico, tipo de financiamiento), misión y visión, objetivos educacionales, resultados de aprendizaje, malla curricular, docentes investigadores/producción científica, trabajos de titulación, docentes en la vinculación, convenios y listado de graduados.

En términos generales la solución de software planteada consistió en dos módulos definidos después de que fuera identificada la información a vincular entre las dos aplicaciones; el primero tuvo el objetivo de exponer la información de interés del SIIU mediante APIs REST; el segundo módulo encargado de actualizar los contenidos del blog, consume los servicios web definidos, procesa los datos recibidos y hace uso de la API REST de WordPress para actualizar el blog.

4.3.1. Objetivos

General

- Integrar el SIIU y blog de la CISIC posibilitando la actualización de contenidos en el blog mediante información proveniente del SIIU.

Específicos

- Configurar los ambientes de desarrollo para el proyecto.
- Exponer la información de interés del SIIU mediante APIs REST.
- Desarrollar un módulo que permita consumir los servicios web basados en REST implementados y haga uso de la API REST de WordPress para actualizar los contenidos informativos del blog.

4.3.2. Arquitectura de integración

La integración de aplicaciones busca su interoperabilidad para que independientemente de plataformas y lenguajes de programación empleados en la implementación puedan comunicarse entre sí y compartan información. La arquitectura propuesta (Figura 31) basada en el uso de servicios web Restful se conforma de tres niveles: en el primero se ubican las aplicaciones a integrar, en el nivel 2 constan los servicios web que exponen la información que será consumida desde otras aplicaciones, y el nivel 3 incluye la funcionalidad para efectuar la tarea requerida.

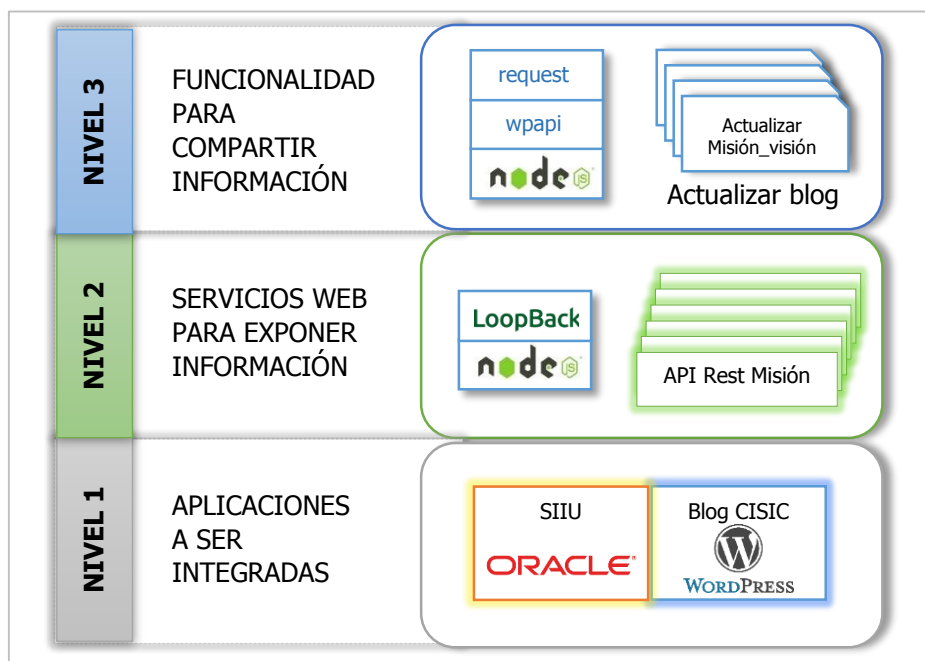


Figura 31. Arquitectura de integración

Fuente: Elaboración propia.

En la Figura 31 constan los siguientes elementos:

- **Nivel 1 (Aplicaciones a ser integradas).**- En este nivel se ubican las aplicaciones que van a ser integradas, en el caso particular de este proyecto son el SIU (base de datos Oracle) y blog de la CISIC (WordPress) del portal web.
- **Nivel 2 (Servicios web para exponer información).**- Mediante servicios web se expone la información que será compartida entre las aplicaciones. En el caso del software para integrar el SIU y blog de carrera, se implementó un conjunto de API's Rest mediante node.js y loopback que permitieron exponer la información de interés proveniente del SIU y que pueden ser invocadas a través de su URL y el método HTTP Get.
- **Nivel 3 (Funcionalidad para compartir información).**- Son las funciones o módulos que permiten compartir información. El objetivo de la integración del SIU y blog de la CISIC es que éste último sea actualizado con información proveniente del SIU; para ello se implementaron varias funciones con node.js y módulos como wpapi y request -entre otros- que invocan las API's Rest según sea el dato que se va a actualizar, llevan a cabo el procesamiento del caso y modifican los contenidos del blog.

La arquitectura propuesta es válida para integrar aplicaciones que requieren compartir información mediante el uso de servicios web, independientemente de las tecnologías empleadas en la implementación de la solución.

4.3.3. Esquema del software

El esquema del software desarrollado es presentado en la Figura 32, en el cual constan los siguientes elementos:

- **Base de datos SIIU.-** Ambiente que alberga la base de datos del SIIU, como parte del ejercicio se restauró únicamente los objetos (tablas, vistas, funciones, entre otros) que contienen la información de interés de los módulos Académico, Biblioteca, Planificación estratégica institucional, Oficina de Relaciones Internacionales, Recursos humanos y Vinculación.
- **Blog de la CISIC.-** Ambiente que contiene el blog de la CISIC en el cual los contenidos seleccionados son actualizados mediante el módulo implementado para tal efecto. Los ítems identificados como actualizables a partir del SIIU fueron definidos después de la revisión de cada elemento tanto en el blog de la CISIC como en los otros 36 blogs de carrera de la UTN.
- **Loopback API Rest.-** Módulo cuyo objetivo es exponer la información de interés proveniente del SIIU. Consiste en un conjunto de APIs REST que reciben parámetros según sea el caso (código de carrera, código del periodo académico) y cuando la solicitud se ha realizado con éxito (HTTP status code 200) devuelven como parte del mensaje de respuesta la información requerida en formato JSON.
- **Funciones para actualización.-** En este módulo se consumen las Loopback APIs REST para obtener la información del SIIU, y se hace uso de la API REST de WordPress para actualizar los contenidos informativos del blog; esta tarea se lleva a cabo cuando la información difiere entre el blog de carrera y el SIIU.

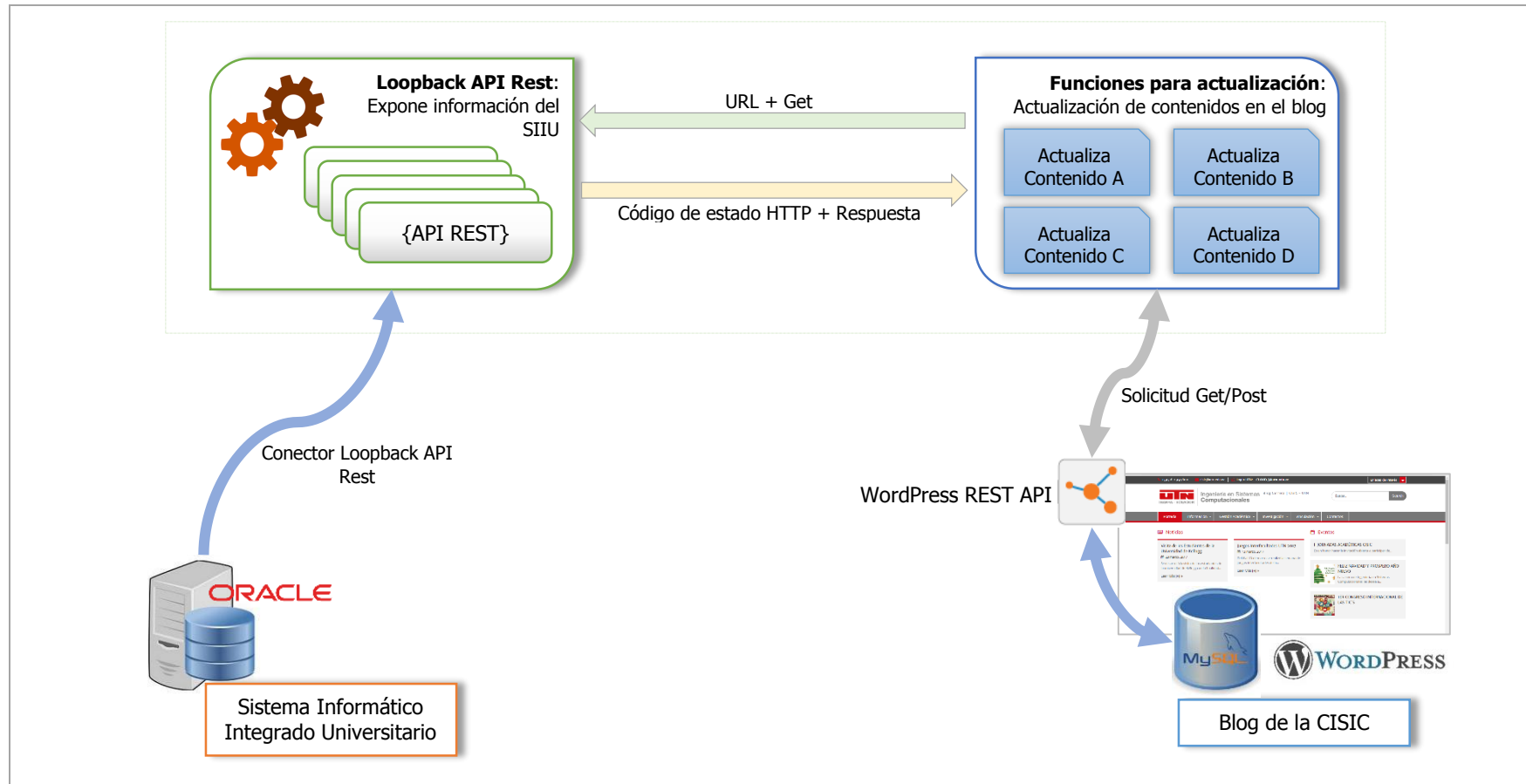


Figura 32. Esquema del software
Fuente: Elaboración propia.

4.4. Pila de producto

La versión final de la pila del producto se presenta en la Tabla 22, incluye los campos *Id* para identificar de manera única a la funcionalidad en cuestión; descripción de la *Funcionalidad* requerida; *Prioridad*; *Estado* define si la tarea está Lista (Ready) pero todavía no inicia, En curso (In progress) o Realizada (Done); *Estimación* indica el número de puntos de historia asignado a cada tarea según su complejidad, se empleó la técnica de póker planning asignando 13 puntos a la tarea más compleja y 5 a la menos compleja; *Sprint* especifica el número de sprint durante el cual la funcionalidad se implementó.

Tabla 22. *Product Backlog*

Id	Funcionalidad requerida	Prioridad	Estado	Estimación	Sprint
1	Actualizar el contenido <i>Misión y visión</i> de la carrera del blog de la CISIC con información proveniente del SIIU	1	Done	8	1
2	Actualizar el contenido <i>Ficha de la carrera</i> del blog de la CISIC con información proveniente del SIIU	2	Done	5	1
3	Actualizar el contenido <i>Objetivos educacionales</i> del blog de la CISIC con información proveniente del SIIU	3	Done	5	2
4	Actualizar el contenido <i>Resultados de aprendizaje</i> del blog de la CISIC con información proveniente del SIIU	4	Done	5	2
5	Actualizar el contenido <i>Malla curricular</i> del blog de la CISIC con información proveniente del SIIU	5	Done	5	2
6	Actualizar el contenido <i>Docentes investigadores / Producción científica</i> del blog de la CISIC con información proveniente del SIIU	6	Done	13	3
7	Actualizar el contenido informativo <i>Docentes en la vinculación</i> del blog de la CISIC	8	Done	5	4
8	Actualizar el contenido informativo <i>Movilidad docente</i> del blog de la CISIC	9	Done	5	4
9	Actualizar el contenido informativo <i>Convenios</i> del blog de la CISIC	10	Done	5	4

Fuente: Elaboración propia.

Se realizó un sprint adicional a los 4 definidos en el product backlog, que comprendió la instalación y configuración de los ambientes necesarios para el desarrollo de los módulos, no brindó una funcionalidad como tal; sin embargo, fue imprescindible para la consecución del resto de sprints.

4.5. Detalle de la ejecución de sprints

Para la ejecución de cada sprint se realizaron las fases de planificación, ejecución y revisión que se explican a continuación.

4.5.1. Planificación del sprint

Las tareas necesarias para completar cada sprint se priorizaron en el backlog estableciendo su orden de ejecución. La Tabla 23 muestra las tareas definidas, éstas se agruparon de acuerdo con las funcionalidades perseguidas, que a su vez se asociaron en base a la categoría del blog con la que estuvieron vinculadas.

Tabla 23. *Funcionalidades definidas para cada Sprint*

Sprint	Id	Tarea	Funcionalidad	Categoría del blog involucrada
1	1.1	Implementación API Rest Misión	F-1.1: Actualizar el contenido informativo <i>Misión y visión</i> de la carrera del blog de la CISIC	Información
	1.2	Implementación API Rest Visión		
	1.3	Implementación funcionalidad para actualización del contenido Misión y Visión de Carrera del blog		
	1.4	Implementación API Rest ConsultasSQL método que devuelve Ficha de carrera	F-1.2: Actualizar el contenido informativo <i>Ficha de la carrera</i> del blog de la CISIC	
	1.5	Implementación funcionalidad para actualización del contenido Ficha de Carrera del blog		
2	2.1	Implementación API Rest Objetivos Educativos	F-2.1: Actualizar el contenido informativo <i>Objetivos educativos</i> del blog de la CISIC	Gestión académica
	2.2	Implementación funcionalidad para actualización del contenido Objetivos Educativos del blog		
	2.3	Implementación API Rest Resultados de aprendizaje	F-2.2: Actualizar el contenido informativo <i>Resultados de aprendizaje</i> del blog de la CISIC	
	2.4	Implementación funcionalidad para actualización del contenido Resultados de aprendizaje del blog		

Sprint	Id	Tarea	Funcionalidad	Categoría del blog involucrada
	2.5	Implementación API Rest ConsultasSQL método que devuelve Malla curricular	F-2.3: Actualizar el contenido informativo <i>Malla curricular</i> del blog de la CISIC	
	2.6	Implementación funcionalidad para actualización del contenido Malla curricular		
3	3.1	Implementación API Rest ConsultasSQL método que devuelve Carga horaria de docentes investigadores	F-3.1: Actualizar el contenido informativo <i>Docentes investigadores/producción científica</i> del blog de la CISIC	Investigación
	3.2	Implementación API Rest ConsultasSQL método que devuelve Producción científica de los docentes de la carrera		
	3.3	Implementación funcionalidad para actualización del contenido Docentes investigadores/Producción científica del blog		
4	4.1	Implementación API Rest ConsultasSQL método que devuelve la carga horaria de Docentes en la vinculación	F-4.1: Actualizar el contenido informativo <i>Docentes en la vinculación</i> del blog de la CISIC	Vinculación
	4.2	Implementación funcionalidad para actualización del contenido Docentes en la vinculación del blog		
	4.3	Implementación API Rest RhuTabAutoridades, método que devuelve la información sobre movilidad docente de la carrera	F-4.2: Actualizar el contenido informativo <i>Movilidad docente</i> del blog de la CISIC	
	4.5	Implementación funcionalidad para actualización del contenido Movilidad Docente del blog		
	4.4	Implementación API Rest ConsultasSQL método que devuelve Convenios de la carrera	F-4.3: Actualizar el contenido informativo <i>Convenios</i> del blog de la CISIC	
	4.6	Implementación funcionalidad para actualización de Convenios del blog		

Fuente: Elaboración propia.

La actualización de un contenido del blog a partir de información del SIIU comprendió en términos generales dos tareas: en la primera se implementó un servicio web basado en Rest que permitió exponer la información seleccionada del SIIU; la segunda incluyó el consumo del servicio web construido además del procesamiento necesario con la API Rest de WordPress para actualizar la información objetivo almacenada en el blog.

4.5.2. Ejecución del sprint

Durante un sprint se implementó varias funcionalidades, el desarrollo de una funcionalidad comprendió la ejecución de varias tareas y siguió una misma lógica general aunque cada una presentó ciertas particularidades. Las tareas especificadas en los sprints backlog comprendieron fases de análisis, codificación y pruebas.

4.5.2.1. Análisis

En cada sprint esta fase inició con la definición de la información a vincular entre las aplicaciones, identificando las tablas de base de datos involucradas. La Tabla 24 contiene el resumen de esta actividad, se incluyó el identificador del sprint, información compartida, categoría del blog a la que pertenece, el módulo del SIIU que la gestiona y las tablas involucradas en cada aplicación.

Tabla 24. *Información a vincular entre las aplicaciones*

Sprint	Información a ser compartida	Blog de la CISIC		Módulo	SIIU
		Corresponde a categoría	Tabla / base de datos		Tabla de la base de datos
1	Ficha de Carrera	Información	_posts	RHU ACA	RHU_TAB_AUTORIDADES ACA_TAB_DEPENDENCIAS ACA_TAB_ESQUEMAS_ESTUDIOS_CICL ACA_TAB_MODALIDADES_ESTUDIOS ACA_TAB_SISTEMAS_ESTUDIOS ACA_TAB_TIPOS_CICLOS_ACADEMICO ACA_TAB_TIPOS_FINANCIAMIENTOS RHU_TAB_CORREO_INSTITUCIONAL
	Misión y visión	Información	_posts	PEI	PEI_TAB_MISION PEI_TAB_VISION
2	Objetivos Educativos	Gestión académica	_posts	ACA	ACA_TAB_OBJ_EDUCACIONALES
	Resultados de Aprendizaje	Gestión académica	_posts	ACA	ACA_TAB_RESULT_APRENDIZAJE
	Malla Curricular	Gestión académica	_posts	ACA	ACA_TAB_MATERIAS_PENSUMS ACA_TAB_MATERIAS ACA_TAB_MATERIAS_PREREQUISITOS
3	Docentes investigadores / Producción Científica	Investigación	_posts	ACA RHU	Carga horaria en investigación ACA_TAB_MATERIAS_PENSUMS ACA_TAB_DISTRIBUTIVOS BI_VIEW_NOMINA_DOCENTE RHU_TAB_PERSONAS

Sprint	Información a ser compartida	Blog de la CISIC Corresponde a categoría	Tabla / base de datos	Módulo	SIIU Tabla de la base de datos
					Produccion científica ACA_TAB_DISTRIBUTIVOS RHU_TAB_OBRAS_PUBLICADAS RHU_TAB_PERSONAS
4	Docentes en la Vinculación	Vinculación	_posts	ACA RHU	ACA_TAB_MATERIAS_PENSUMS ACA_TAB_DISTRIBUTIVOS BI_VIEW_NOMINA_DOCENTE RHU_TAB_PERSONAS
	Movilidad Docente	Vinculación	_posts	ORI	ORI_MOVILIDAD ORI_MOVILIDAD_PARTICIPANTES ORI_CONVENIO
	Convenios	Vinculación	_posts	ORI VIN	ORI_CONVENIO ORI_TIPO_CONVENIO ORI_CONVENIO_INSTITUCION ORI_CONVENIO_DEPENDENCIAS VIN_TAB_INSTITUCION

Nota: Leyenda de la tabla (Académico - ACA, Planificación estratégica institucional - PEI, Oficina de Relaciones Internacionales - ORI, Recursos humanos - RHU y Vinculación – VIN)

Fuente: Elaboración propia.

Como se explicó anteriormente el software desarrollado comprendió dos módulos, el primero para exponer la información del SIIU (módulo A) y el segundo para actualizar contenidos seleccionados en el blog (módulo B); debido a ello en la fase de análisis de cada sprint se definió qué API's Rest del módulo A y cuáles funciones del módulo B con sus respectivos parámetros se implementarían.

4.5.2.2. Codificación

Las API Rest del módulo de exposición de información y las funciones del módulo de actualización de contenidos fueron desarrollados con Node.js. En la sección izquierda de la Figura 33 se presenta parte de la codificación para la API Rest Misión, ésta indica la inclusión del método fidTIC que recibe el parámetro centroCosCodigo de tipo string, y devuelve en caso exitoso (HTTP status code 200) un array; además se incluye la URL de acceso y el método HTTP; en la sección derecha de la figura consta parte del archivo empleado para el mapeo de los modelos de datos de Oracle y JavaScript.

<pre> module.exports = function(PeiTabMision) { PeiTabMision.findTIC = function(value,cb){ PeiTabMision.find({ where :{ centroCosCodigo : value, estado: 'A' } }, cb); }; PeiTabMision.remoteMethod("findTIC", { description: "Mision", accepts:{ arg: "centroCosCodigo", type: "string", }, returns: { arg: "misiones", type: "array" }, http: { path: "/findTIC", verb: "get" } }); </pre>	<pre> { "name": "PeiTabMision", "options": { "idInjection": false, "oracle": { "schema": "UTNDB", "table": "PEI_TAB_MISION" }, "relations": { } }, "properties": { "codigo": { "type": "String", "required": true, "length": 40, "precision": null, "scale": null, "id": 1, "oracle": { "columnName": "CODIGO", "dataType": "VARCHAR2", "dataLength": 40, "dataPrecision": null, "dataScale": null, "nullable": "N" } }, "descripcion": { "type": "String", "required": true, "length": 4000, "precision": null, "scale": null, "oracle": { "columnName": "DESCRIPCION", </pre>
---	---

Figura 33. Código de API Rest Misión

Fuente: Elaboración propia.

La Tabla 25 contiene el listado de endpoints para las API's Rest que exponen la información del SIIU; se incluye la descripción de la información expuesta, URL de solicitud, método HTTP, el modelo en formato JSON de los datos que retorna en caso de éxito, y los parámetros recibidos.

Tabla 25. API's Rest implementadas

Sprint	Descripción	Request URL	Método	Respuesta en HTTP status code 200	Parámetros Descripción (Tipo de dato)
1	Misión de carrera	/PeiTabMisions/findTIC	GET	{ "misiones": [{ }] }	<i>centroCosCodigo</i> - Código de carrera (string)
	Visión de carrera	/PeiTabVisions/findTIC	GET	{ "visiones": [{ }] }	<i>centroCosCodigo</i> - Código de carrera (string)
	Ficha de carrera	/ConsultasSQLs/findFC	GET	{ "autoridades": [{ }] }	<i>ciclo</i> - Código de ciclo académico (string) <i>codigo</i> - Código de carrera (string)
2	Objetivos educativos	/AcaTabObjEducativos/findTIC	GET	{ "objetivosEd": [{ }] }	<i>depenCodigo</i> - Código de carrera (string)
	Resultados de aprendizaje	/AcaTabResultAprendizajes/findTIC	GET	{ "resultadosApre": [{ }] }	<i>depenCodigo</i> - Código de carrera (string)
	Malla curricular	/ConsultasSQLs/findMC	GET	{ "malla": [{ }] }	<i>ciclo</i> - Código de ciclo académico (string) <i>dependencia</i> - Código de carrera (string)

Sprint	Descripción	Request URL	Método	Respuesta en HTTP status code 200	Parámetros Descripción (Tipo de dato)
3	Docentes investigadores	/ConsultasSQLs/findDI	GET	{ "docentes": [{}] }	<i>ciclo</i> - Código de ciclo académico (string) <i>codigo</i> - Código de carrera (string)
	Producción científica de docentes	/ConsultasSQLs/findPC	GET	{ "producciones": [{}] }	<i>ciclo</i> - Código de ciclo académico (string) <i>codigo</i> - Código de carrera (string)
4	Docentes en la vinculación	/ConsultasSQLs/findDV	GET	{ "docentes": [{}] }	<i>ciclo</i> - Código de ciclo académico (string) <i>dependencia</i> - Código de carrera (string)
	Movilidad docente	/ConsultasSQLs/findMD	GET	{ "movilidades": [{}] }	<i>dependencia</i> - Código de carrera (string)
	Convenios	/ConsultasSQLs/findC	GET	{ "convenios": [{}] }	<i>dependencia</i> - Código de carrera (string)

Fuente: Elaboración propia.

La Tabla 26 describe las funciones implementadas en el módulo encaminado a la actualización de contenidos del blog de la CISIC, se describe la categoría del blog cuyo contenido es actualizado mediante la función, el nombre de la misma, así como las API's Rest que consume.

Tabla 26. Funciones implementadas en el módulo de actualización de información

Sprint	Categoría del blog	Nombre de la función	Descripción	API's Rest que consume
1	Información	misionVision.consultarMV	Actualiza el contenido <i>Misión y Visión</i> de la carrera en el blog de la CISIC	/PeiTabMisions/findTIC /PeiTabVisions/findTIC
		fichaCarrera.consultarFC	Actualiza el contenido <i>Ficha de la carrera</i> en el blog de la CISIC	/ConsultasSQLs/findFC
2	Gestión académica	objetivosEducativos.consultarOE	Actualiza el contenido <i>Objetivos educativos</i> en el blog de la CISIC	/AcaTabObjEducativos/findTIC
		resultadosAprendizaje.consultarRA	Actualiza el contenido <i>Resultados de aprendizaje</i> en el blog de la CISIC	/AcaTabResultAprendizajes/findTIC
		mallaCurricular.consultarMallaCurricular	Actualiza el contenido <i>Malla curricular</i> en el blog de la CISIC	/ConsultasSQLs/findMC
3	Investigación	docentesInvestigadores.consultarDI	Actualiza el contenido <i>Docentes investigadores</i> en el blog de la CISIC	/ConsultasSQLs/findDI
			Actualiza el contenido <i>Producción científica</i> de docentes en el blog de la CISIC	/ConsultasSQLs/findPC
4	Vinculación	docentesVinculacion.consultarDV	Actualiza el contenido <i>Docentes en la vinculación</i> en el blog de la CISIC	/ConsultasSQLs/findDV
		movilidadDocente.consultarMovilidadD	Actualiza el contenido <i>Movilidad docente</i> en el blog de la CISIC	/ConsultasSQLs/findMD
		convenios.consultarConvenios	Actualiza el contenido <i>Convenios</i> en el blog de la CISIC	/ConsultasSQLs/findC

Fuente: Elaboración propia.

En la Figura 34 se muestra parte del código de la función encargada de actualizar la misión y visión en el blog, en las primeras líneas consta la invocación a las API Rest para recuperar la información del SIIU; después, la lógica que permite procesar la edición del post mediante la API Rest de WordPress.

```

var request = require("request");
require('dotenv').load();
var wp = require("./wpapiLoader");
function consultarOracleMV(codigo) {
  var resultados= [];
  var misiondb= "";
  var visiondb="";
  var misionweb="";
  var visionweb="";
  var URLMision = process.env.hostApi + "/api/PeiTabMisiones/findTIC?centroCosCodigo=00165";
  var webMision = process.env.endpoint + "/wp/v2/pages/10"
  var URLVision = process.env.hostApi + "/api/PeiTabVisions/findTIC?centroCosCodigo=00165";
  request({url: URLVision,json: true }, function (error, response, body) {
  if (!error && response.statusCode === 200) {
    visiondb = body.visiones[0].descripcion;
    request({url: URLMision,json: true }, function (error, response, body) {
      if (!error && response.statusCode === 200) {
        misiondb= body.misiones[0].descripcion;
        request({url: webMision,json: true }, function (error, response, body) {
          if (!error && response.statusCode === 200) {
            misionweb = body.content.rendered;
            console.log("Consultando Mision y vision ");
            if (misionweb.includes(misiondb) && misionweb.includes(visiondb) ) {
              console.log("Misión y visión están actualizados");
            }else {
              console.log("Necesita actualizar la mision y vision");
              var inicioMision = misionweb.indexOf("MISIÓN") + 23;
              var finMision = misionweb.indexOf("VISIÓN") - 44;
              var inicioVision = misionweb.indexOf("VISIÓN") + 23 ;
              var finVision= misionweb.indexOf("&nbsp;") - 8;
              var misionReemplazar = misionweb.substring(inicioMision, finMision);
              var visionReemplazar = misionweb.substring(inicioVision,finVision);
              var misionVisionNueva = misionweb.replace(misionReemplazar,misiondb).replace(visionReemplazar,visiondb);
            }
          }
        }
      }
    }
  }
}

```

Figura 34. Función que actualiza el contenido Misión y visión

Fuente: Elaboración propia.

A continuación se explica la actualización del contenido del blog *Misión y visión* perteneciente a la categoría *Información*, en la Figura 35 está seleccionado dicho elemento; el tipo de información es texto, los datos se encuentran disponibles en el SIIU y si es posible extraerlos para actualizar la publicación del blog (ver Tabla 21). El módulo del SIIU que gestiona la visión y misión de las dependencias de la universidad es Planificación Estratégica Institucional (PEI), por tanto una vez que fueron identificadas las tablas que almacenan dicha información se verificó el script SQL respectivo que permite obtenerla.

+593 6 2 997800 | cisic@utn.edu.ec | Login UTN: CLOUD /@utn.edu.ec

Blog Carrera | CISIC - UTN

Portada Información ▼ Gestión Académica ▼ Investigación ▼ Vinculación ▼

/ Información / Misión y Visión

Nuestra Carrera

Ficha de Carrera

Historia de la carrera

Misión y Visión

MISIÓN

La Carrera de Ingeniería en Sistemas Computacionales forma ingenieros competentes, críticos, humanistas, líderes y emprendedores con responsabilidad social; genera, fomenta y ejecuta procesos tecnológicos, de conocimientos científicos y de innovación en el desarrollo de soluciones informáticas.

VISIÓN

La Carrera de Ingeniería en Sistemas Computacionales, en el año 2020, será un referente regional y nacional.

Figura 35. Contenido Misión y visión del blog
Fuente: Ambiente de desarrollo configurado.

La Figura 36 muestra el resultado de invocar las API's Rest Misión y Visión mediante su URL, proporcionando el parámetro respectivo en cada caso; PeiTabMisions devuelve la misión y PeiTabVisions expone la visión almacenada en el SIIU. Los datos publicados en el blog (Figura 35) difieren de los devueltos por las API Rest (Figura 36), en ese caso amerita la actualización en el blog entonces la función `misionVision.consultarMV` realiza dicha tarea.

GET /PeiTabMisions/findTIC Mision

Response Body

```

{
  "visiones": [
    {
      "codigo": "00010",
      "descripcion": "La Carrera de Ingeniería en Sistemas Computacionales, en el año 2020, será un referente regional y nacio
      "fechaElaboracion": "2014-09-17T05:00:00.000Z",
      "estado": "A",
      "observacion": null,
      "pediAnioFin": "2017",
      "pediAnioInicio": "2013",
      "centroCosCodigo": "00165",
      "visionCodigo": "00003",
      "visionAnioInicio": "2013",
      "visionAnioFin": "2017",
      "visionCentroCosCodigo": "00005"
    }
  ]
}

```

Response Code

200

GET /PeiTabVisions/findTIC Vision

Response Body

```

{
  "visiones": [
    {
      "codigo": "00010",
      "descripcion": "La Carrera de Ingeniería en Sistemas Computacionales, en el año 2020, será un referente regional y nacio
      "fechaElaboracion": "2014-09-17T05:00:00.000Z",
      "estado": "A",
      "observacion": null,
      "pediAnioFin": "2017",
      "pediAnioInicio": "2013",
      "centroCosCodigo": "00165",
      "visionCodigo": "00003",
      "visionAnioInicio": "2013",
      "visionAnioFin": "2017",
      "visionCentroCosCodigo": "00005"
    }
  ]
}

```

Response Code

200

Figura 36. Datos devueltos por las API's Rest Misión y visión

Fuente: Loopback API Explorer.

Después de efectuar la actualización en el blog, la misión y visión publicadas en éste concuerdan con las alojadas en el SIIU; la Figura 37 muestra esa tarea: (1) los mensajes mostrados en consola por **consultarMV** indican que se requiere llevar un update de datos

en el blog; (2) la información es actualizada; (3) los mensajes presentados en consola advierten que la información ya está actualizada.


1	Mensaje presentado en consola por la función de actualización (antes de cumplir su objetivo)	<pre>consultando Mision y vision necesita actualizar la mision y vision</pre>
2	<p style="text-align: center;">CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES</p>  <p style="text-align: center;">MISIÓN</p> <p>La Carrera de Ingeniería en Sistemas Computacionales forma ingenieros competentes, críticos, humanistas, líderes y emprendedores con responsabilidad social; genera, fomenta y ejecuta procesos tecnológicos, de conocimientos científicos y de innovación en el desarrollo de soluciones informáticas; se vincula con el medio con criterios de sustentabilidad para contribuir al desarrollo social, económico, cultural y ecológico de la región y el país.</p> <p style="text-align: center;">VISIÓN</p> <p>La Carrera de Ingeniería en Sistemas Computacionales, en el año 2020, será un referente regional y nacional en la formación de ingenieros en sistemas computacionales.</p>	
3	Mensaje presentado en consola por la función de actualización (una vez ha cumplido su tarea)	<pre>consultando Mision y vision consultando Mision y vision</pre>

Figura 37. Misión y visión de la CISIC almacenadas en el SIIU

Fuente: Elaboración propia.

4.5.2.3. Pruebas

Cada funcionalidad implementada fue sometida a pruebas, los casos de prueba aplicados se detallan en la sección 4.7. Una vez que todas las tareas definidas fueron completadas, se procedió a verificar el cumplimiento del resultado esperado en cada sprint (Sección 4.8).

4.6. Sprints backlogs

Los sprint backlogs finales con las tareas necesarias para cumplir lo requerido contienen la siguiente información:

- **Datos generales del sprint.**- Id, nombre y descripción del sprint.
- **Resultado.**- Indica el incremento del producto que debe ser alcanzado durante cada sprint.
- **Equipo de desarrollo.**- Número de equipos de desarrollo participantes.
- **Días de duración del sprint.**- Especifica la cantidad de días laborables durante los cuales se efectúa el sprint.
- **Fecha inicio/Fecha fin.**- Establecen las fechas de inicio y finalización del sprint.
- **Número óptimo de horas de trabajo por persona.**- Es un valor que muestra el número de horas óptimo de trabajo para cada individuo del equipo de trabajo; dato obtenido en base a la multiplicación del número de días de duración del sprint por las horas de trabajo diario (6 h), del total resultante se calculó el 80% debido a que es casi imposible que un desarrollador dedique la totalidad de su tiempo a las tareas definidas en un sprint ya que comúnmente surgen imprevistos.
- **Número de horas de trabajo del equipo.**- Constituye el número total de horas de trabajo del equipo destinadas al sprint.
- **Información de las tareas del sprint.**- Es una tabla que contiene el id de la tarea, descripción, estado (No iniciada – NI, En curso – EC, Completada – CO), fecha de inicio y finalización, y número de horas empleadas para su culminación.
- **Tabla burndown del sprint.**- Presenta el tiempo restante o trabajo pendiente expresado en horas que falta realizar para completar el sprint. En los backlogs de cada sprint se incluye el tiempo restante real (TR) y el tiempo restante estimado (TE). Para calcular el primero, al tiempo restante total del día anterior (o tiempo estimado total en el caso de que el cálculo se realice para el primer día del sprint) se restó el total de horas de trabajo llevado a cabo en el día. El tiempo restante promedio se calculó substrayendo del tiempo restante total del día anterior el promedio de horas de trabajo diario, valor obtenido a partir de la división del número total de horas planificadas para el número de días de duración del sprint.

- **Gráfico burndown del sprint.**- Es la representación gráfica del trabajo que queda pendiente para concluir el sprint; el eje X constituye el tiempo (días del sprint) mientras que el eje Y es el trabajo pendiente.

4.6.1. Sprint 0

Nombre

Configuración y actualización de ambientes

Descripción

El sprint 0 permitió la configuración de los ambientes requeridos para el desarrollo del proyecto.

Resultado

Al finalizar el Sprint 0 se encontraron listos los siguientes ambientes:

- a) *Base de datos SIIU.*- Sistema operativo Centos, alberga Oracle XE que contiene parte de la base de datos del SIIU, en particular información de interés perteneciente a los módulos Académico, Biblioteca, Planificación estratégica institucional, Oficina de Relaciones Internacionales, Recursos humanos y Vinculación.
- b) *Blog de la Carrera de Ingeniería en Sistemas Computacionales.*- CMS Wordpress 4.7, base de datos MySQL 5.6.
- c) *Node.js.*- Ambiente de ejecución para JavaScript, instalación de módulo para base de datos Oracle, loopback, underscore, dotenv, wpapi, entre otros.

Equipo de desarrollo: 1

Días de duración del sprint: 15

Fecha inicio: 01/05/2017

Fecha fin: 19/05/2017

Número óptimo de horas de trabajo por persona (80%): 72

Número de horas de trabajo del equipo: 72

El sprint 0 se centró en la instalación y configuración de los ambientes necesarios para el desarrollo del proyecto, se realizaron 18 tareas (Tabla 27); este sprint no brinda funcionalidad pero es necesario para llevar a cabo el resto de sprints.

Tabla 27. Backlog de Sprint 0

Id	Tarea	Estado	Fecha inicio	Fecha fin	Horas estimadas	Trabajo realizado por día (h)															
						01/05/2017	02/05/2017	03/05/2017	04/05/2017	05/05/2017	08/05/2017	09/05/2017	10/05/2017	11/05/2017	12/05/2017	15/05/2017	16/05/2017	17/05/2017	18/05/2017	19/05/2017	
0.1	Instalación VM Centos (Oracle)	CO	1-may.	1-may.	3	3															
0.2	Instalación y configuración de Oracle XE	CO	1-may.	1-may.	2	2															
0.3	Creación de esquema y carga de datos Módulo FIN	CO	2-may.	2-may.	1		1														
0.4	Creación de esquema Módulo ACA	CO	3-may.	3-may.	3			3													
0.5	Carga de información Módulo ACA	CO	4-may.	10-may.	20				4	4	4	4	4								
0.6	Creación de esquema Módulo PEI	CO	11-may.	11-may.	3								3								
0.7	Carga de información Módulo PEI	CO	12-may.	15-may.	9									5	4						
0.8	Creación de esquema Módulo RHU	CO	2-may.	2-may.	1		1														
0.9	Carga de información Módulo RHU	CO	2-may.	2-may.	2		2														
0.10	Creación de esquema Módulo ORI	CO	11-may.	11-may.	1								1								
0.11	Carga de información Módulo ORI	CO	12-may.	15-may.	3									1	2						
0.12	Creación de esquema Módulo MOV	CO	11-may.	11-may.	1													1			
0.13	Carga de información Módulo MOV	CO	17-may.	17-may.	3													3			
0.14	Creación de esquema Módulo BIB	CO	17-may.	17-may.	1													1			
0.15	Carga de información Módulo BIB	CO	16-may.	16-may.	4														4		
0.16	Instalación de Apache, MySQL y Wordpress	CO	18-may.	18-may.	4														4		
0.17	Restaurar BD para el blog de la CISIC	CO	19-may.	19-may.	1														1		
0.18	Instalación de Node-oracledb	CO	19-may.	19-may.	5														5		
Total					67	5	4	3	4	4	4	4	4	4	4	6	6	5	4	4	6
Burndown Sprint 0: Configuración y actualización de ambientes																					
Tiempo restante real					62	58	55	51	47	43	39	35	31	25	19	14	10	6	0		
Tiempo restante estimado					62,5	58,1	53,6	49,1	44,7	40,2	35,7	31,3	26,8	22,3	17,9	13,4	8,9	4,5	0,0		

Nota: Estado CO = Completada

Fuente: El diseño de la tabla fue adaptado a partir de template propuesto por (Banks, 2007)

La Figura 38 es el gráfico burndown del Sprint 0, indica el trabajo pendiente expresado en horas por cada día; la línea azul corresponde a la estimación ideal mientras que la fucsia al trabajo restante real. El gráfico se construyó en base a la información ubicada al final de la Tabla 27, identificada por el subtítulo “Burndown Sprint 0: Configuración y actualización de ambientes”.

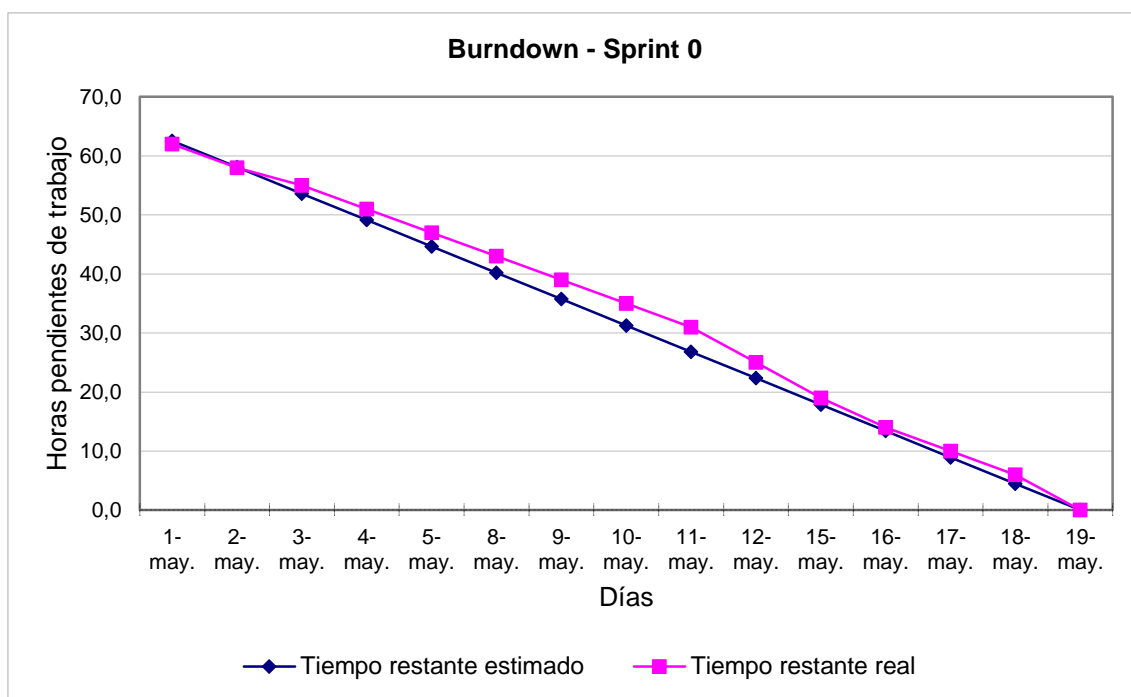


Figura 38. Burndown de Sprint 0

Fuente: Elaboración propia.

El sprint 0 se cumplió en el plazo estimado, al finalizar la primera semana quedaban pendientes 47 horas para completar el sprint, al finalizar la segunda semana 25 horas, y a la fecha planificada de finalización del sprint se lograron cumplir todas las tareas definidas.

4.6.2. Sprint 1

Nombre

Vinculación de información correspondiente a la categoría Información del blog de carrera

Descripción

El primer sprint tuvo como objetivo la actualización de contenidos informativos seleccionados pertenecientes a la categoría Información del blog.

Resultado

Los contenidos *Ficha de carrera*, y *misión y visión de la carrera* se actualizan en el blog de la CISIC, cuando la información publicada no coincide con la almacenada en el SIIU.

Equipo de desarrollo: 1

Días de duración del sprint: 15

Fecha inicio: 22/05/2017

Fecha fin: 09/06/2017

Número óptimo de horas de trabajo por persona (80%): 72

Número de horas de trabajo del equipo: 72

El sprint 1 permitió la actualización de contenidos pertenecientes a la categoría Información del blog de la CISIC, fue llevado a cabo durante 3 semanas mediante la ejecución de 5 tareas especificadas en la Tabla 28, donde además se indica las fechas de inicio y finalización de cada una, así como el tiempo empleado expresado en horas.

Los ítems que pueden ser actualizados en el blog como resultado del sprint 1 son la Ficha de carrera, y Misión y visión. Las tareas del sprint incluyeron la exposición de información del SIIU mediante APIs Rest correspondiente a los ítems mencionados, y la actualización a través de métodos que consumen los servicios implementados y se encargan de modificar los contenidos referidos en el blog.

Tabla 28. Backlog de Sprint 1

Id	Tarea	Estado	Fecha inicio	Fecha fin	Horas estimadas	Trabajo realizado por día (h)														
						22/05/2017	23/05/2017	24/05/2017	25/05/2017	26/05/2017	29/05/2017	30/05/2017	31/05/2017	01/06/2017	02/06/2017	05/06/2017	06/06/2017	07/06/2017	08/06/2017	09/06/2017
1.1	Implementación API Rest Misión (análisis, codificación, pruebas)	CO	22-may.	24-may.	9	3	3	3												
1.2	Implementación API Rest Visión	CO	25-may.	29-may.	9				3	3	3									
1.3	Implementación funcionalidad para actualización del contenido Misión y Visión de Carrera del blog	CO	29-may.	9-jun.	18							4	4	4	4	2				
1.4	Implementación API Rest ConsultasSQL método que devuelve Ficha de carrera	CO	2-ene.	2-jun.	9	1	1	1	2	2	2									
1.5	Implementación funcionalidad para actualización del contenido Ficha de Carrera del blog	CO	5-jun.	9-jun.	18											4	5	3	6	
Total					63	4	4	4	5	5	5	4	4	4	4	2	4	5	3	6
Burndown Sprint 1: Vincular la información de la categoría Información del blog de carrera																				
Tiempo restante real						59,0	55,0	51,0	46,0	41,0	36,0	32,0	28,0	24,0	20,0	18,0	14,0	9,0	6,0	0,0
Tiempo restante estimado						58,8	54,6	50,4	46,2	42,0	37,8	33,6	29,4	25,2	21,0	16,8	12,6	8,4	4,2	0,0

Nota: Estado CO = Completada

Fuente: El diseño de la tabla fue adaptado a partir de template propuesto por (Banks, 2007)

La Figura 39 presenta el gráfico burndown del Sprint 1; el número de horas de trabajo completadas en la primera, segunda y tercera semana fueron 22 h, 21 h y 20 h respectivamente; logrando el cumplimiento de la totalidad de actividades planificadas durante el tiempo destinado al sprint.

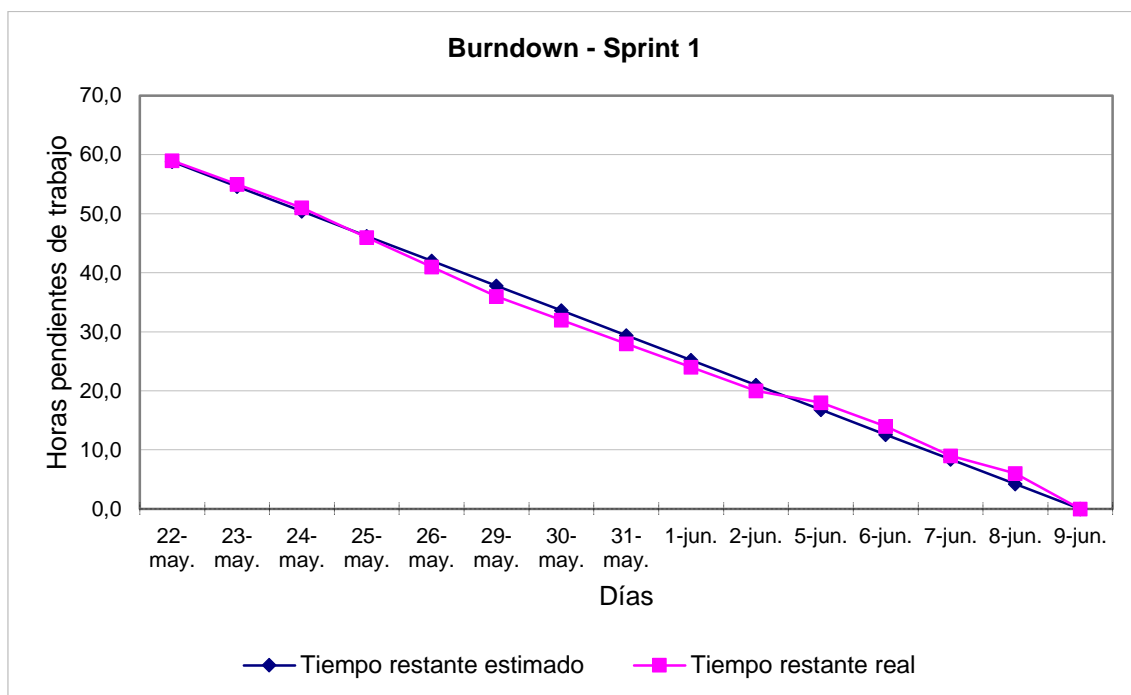


Figura 39. Burndown de Sprint 1

Fuente: Elaboración propia.

En el gráfico burndown el eje X representa a los días durante los cuales se llevó a cabo el sprint 1, mientras que el eje Y a las horas de trabajo restantes para completarlo; la línea de color fucsia (tiempo restante real) está muy cercana a la línea de color azul (tiempo restante estimado) lo que indica que el sprint fue llevado a cabo con un avance muy cercano al tiempo ideal.

4.6.3. Sprint 2

Nombre

Vinculación de información correspondiente a la categoría Gestión Académica del blog de carrera

Descripción

El segundo sprint tuvo como finalidad la actualización de contenidos informativos seleccionados pertenecientes a la categoría Gestión Académica del blog.

Resultado

Los contenidos *Objetivos Educativos*, *Resultados de Aprendizaje* y *Malla Curricular* se actualizan en el blog de la CISIC, cuando la información publicada no coincide con la almacenada en el SIIU.

Equipo de desarrollo: 1

Días de duración del sprint: 15

Fecha inicio: 12/06/2017

Fecha fin: 30/06/2017

Número óptimo de horas de trabajo por persona (80%): 72

Número de horas de trabajo del equipo: 72

Para alcanzar el resultado del sprint 2 se definieron las tareas descritas en la Tabla 29, de manera similar al sprint anterior se implementaron servicios web basados en Rest para exponer la información del SIIU (objetivos educativos, resultados de aprendizaje, malla curricular), y métodos que permiten actualizar los ítems señalados pertenecientes a la categoría Gestión Académica en el blog de la CISIC.

Las tareas correspondientes al sprint 2 se completaron en los quince días de duración para las que fue planificado; al quinto día del sprint se habían efectuado 20 horas de trabajo, para el día décimo 40 horas de trabajo, mientras que en el quinceavo día se cumplieron las 70 h definidas para el sprint.

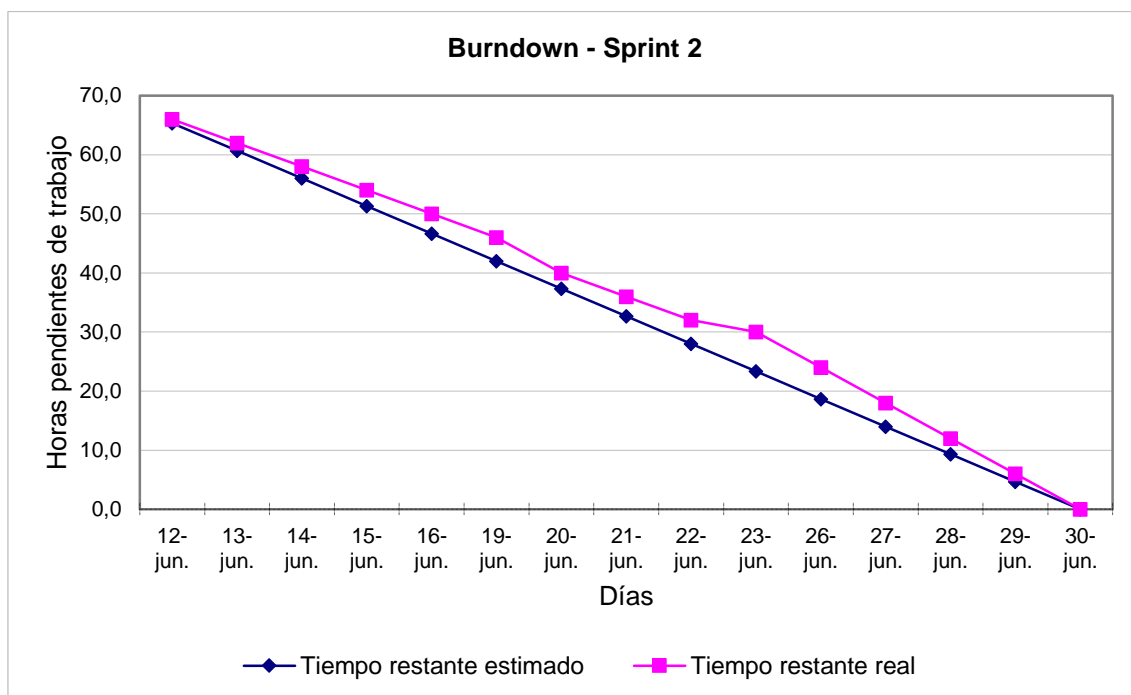


Figura 40. Burndown de Sprint 2

Fuente: Elaboración propia.

En la gráfica burndown del sprint 2 (Figura 40) la línea del tiempo restante real (color fucsia) está por encima de la línea del tiempo restante estimado (color azul) lo que indica que el sprint fue desarrollado a menor velocidad de lo estimado, pero se completó durante el tiempo establecido. Al final de la primera semana el tiempo restante real fue de 50 h mientras que el estimado de 46,7 h; al acabar la segunda semana el tiempo restante real fue de 30 h mientras que el estimado de 23,3 h; al finalizar la tercera semana se completaron todas las horas de trabajo pendientes del sprint.

4.6.4. Sprint 3

Nombre

Vinculación de información correspondiente a la categoría Investigación del blog de carrera

Descripción

El tercer sprint tuvo como finalidad la actualización de contenidos informativos seleccionados pertenecientes a la categoría Investigación del blog.

Resultado

Los contenidos *Docentes investigadores* y *Producción Científica* se actualizan en el blog de la CISIC cuando la información publicada no coincide con la almacenada en el SIIU.

Equipo de desarrollo: 1

Días de duración del sprint: 10

Fecha inicio: 03/07/2017

Fecha fin: 21/07/2017

Número óptimo de horas de trabajo por persona (80%): 72

Número de horas de trabajo del equipo: 72

En la Tabla 30 constan las tareas del sprint 3 definidas para que los contenidos informativos *Docentes investigadores* y *Producción Científica* sean actualizados con información del SIIU. Las tareas relacionadas con la exposición de los datos de interés del SIIU se implementaron primero ya que fueron empleadas por las tareas creadas para modificar los contenidos.

Tabla 30. Backlog de Sprint 3

Id	Tarea	Estado	Fecha inicio	Fecha fin	Horas estimadas	Trabajo realizado por día (h)														
						03/07/2017	04/07/2017	05/07/2017	06/07/2017	07/07/2017	10/07/2017	11/07/2017	12/07/2017	13/07/2017	14/07/2017	17/07/2017	18/07/2017	19/07/2017	20/07/2017	21/07/2017
3.1	Implementación API Rest ConsultasSQL método que devuelve Carga horaria de docentes investigadores	CO	3-jul.	6-jul.	12	3	3	3	3											
3.2	Implementación API Rest ConsultasSQL método que devuelve Producción científica de los docentes de la carrera	CO	4-jul.	10-jul.	12		2	2	3	3	2									
3.3	Implementación funcionalidad para actualización del contenido Docentes investigadores/Producción científica del blog	CO	11-jul.	21-jul.	36							5	5	3	4	4	3	3	4	5
Total					60	3	5	5	6	3	2	5	5	3	4	4	3	3	4	5
Burndown Sprint 3: Vincular la información de la categoría Investigación del blog de carrera																				
Tiempo restante real						57,0	52,0	47,0	41,0	38,0	36,0	31,0	26,0	23,0	19,0	15,0	12,0	9,0	5,0	0,0
Tiempo restante estimado						56,0	52,0	48,0	44,0	40,0	36,0	32,0	28,0	24,0	20,0	16,0	12,0	8,0	4,0	0,0

Nota: Estado CO = Completada

Fuente: El diseño de la tabla fue adaptado a partir de template propuesto por (Banks, 2007)

La Figura 41 representa el burndown correspondiente al Sprint 3, la línea del tiempo restante real (color fucsia) está muy cercana a la línea del tiempo restante estimado (color azul); lo que indica que el avance del sprint se llevó a cabo casi de acuerdo a lo estimado logrando cumplirse en el plazo establecido.

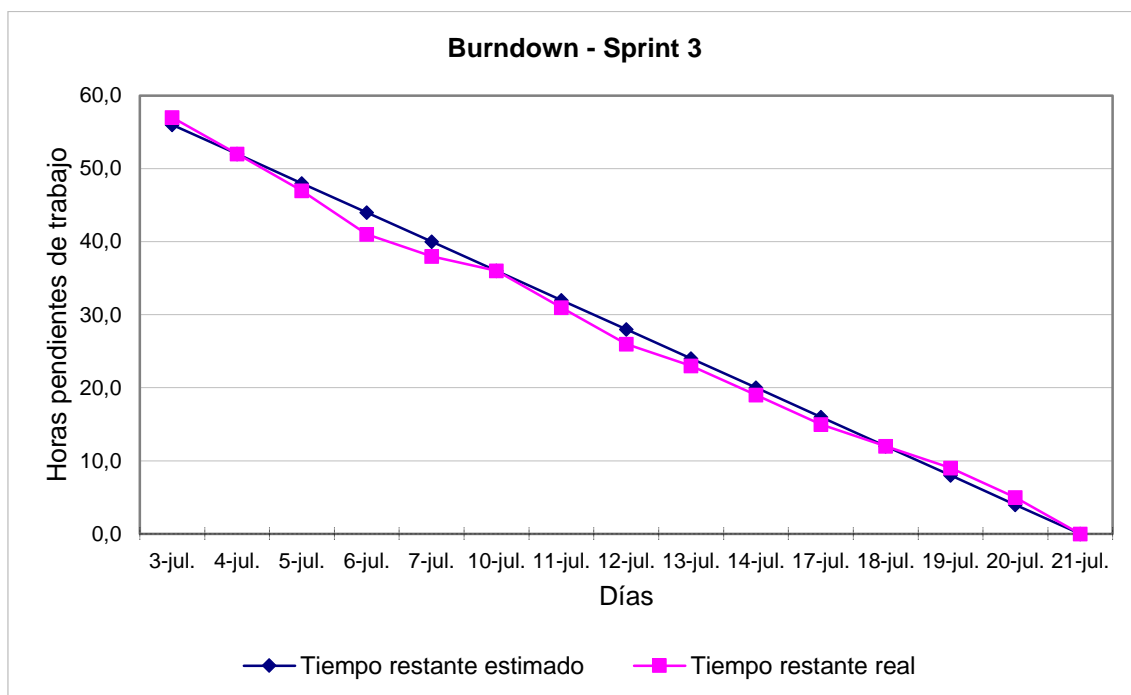


Figura 41. Burndown de Sprint 3

Fuente: Elaboración propia.

Al comparar el tiempo restante real y estimado para varios días se puede apreciar que el sprint fue ejecutado a un buen ritmo de trabajo; por ejemplo para el día 3 quedaban pendiente 47h reales versus las 48h estimadas; día 6 36h reales / 36h estimadas; día 9 23h reales / 24h estimadas y al día 12 12h reales / 12h estimadas.

4.6.5. Sprint 4

Nombre

Vincular la información de la categoría Vinculación del blog de carrera

Descripción

Sprint definido para actualizar los contenidos informativos seleccionados pertenecientes a la categoría Vinculación del blog.

Resultado

Los contenidos *Docentes en la Vinculación, Movilidad Docente y Convenios* se actualizan en el blog de la CISIC cuando la información publicada no coincide con la almacenada en el SIIU.

Equipo de desarrollo: 1

Días de duración del sprint: 15

Fecha inicio: 24/07/2017

Fecha fin: 04/08/2017

Número óptimo de horas de trabajo por persona (80%): 72

Número de horas de trabajo del equipo: 72

La Tabla 31 contiene el listado de las tareas establecidas para cumplir el sprint 4, enfocado en la actualización de contenidos de la categoría Vinculación. Las tareas planteadas se desarrollaron durante 3 semanas, alcanzando el resultado esperado para el sprint en el plazo planificado.

Tabla 31. Backlog de Sprint 4

Id	Tarea	Estado	Fecha inicio	Fecha fin	Horas estimadas	Trabajo realizado por día (h)														
						24/07/2017	25/07/2017	26/07/2017	27/07/2017	28/07/2017	29/07/2017	30/07/2017	31/07/2017	01/08/2017	02/08/2017	31/07/2017	01/08/2017	02/08/2017	03/08/2017	04/08/2017
4.1	Implementación API Rest ConsultasSQL método que devuelve la carga horaria de Docentes en la vinculación	CO	24-jul.	25-jul.	10	5	5													
4.2	Implementación funcionalidad para actualización del contenido Docentes en la vinculación del blog	CO	26-jul.	29-jul.	15			5	3	4	3									
4.3	Implementación API Rest ConsultasSQL método que devuelve la información sobre movilidad docente de la carrera	CO	30-jul.	31-jul.	10							5	5							
4.4	Implementación API Rest ConsultasSQL método que devuelve Convenios de la carrera	CO	1-ago.	2-ago.	10								5	5						
4.5	Implementación funcionalidad para actualización del contenido Movilidad Docente del blog	CO	31-jul.	1-ago.	10										5	5				
4.6	Implementación funcionalidad para actualización del contenido Convenios del blog	CO	2-ago.	4-ago.	15												5	5	5	
Total					70	5	5	5	3	4	3	5	5	5	5	5	5	5	5	5
Burndown Sprint 4: Vincular la información de la categoría Vinculación del blog de carrera																				
Tiempo restante real						65,0	60,0	55,0	52,0	48,0	45,0	40,0	35,0	30,0	25,0	20,0	15,0	10,0	5,0	0,0
Tiempo restante estimado						65,3	60,7	56,0	51,3	46,7	42,0	37,3	32,7	28,0	23,3	18,7	14,0	9,3	4,7	0,0

Nota: Estado CO = Completada

Fuente: El diseño de la tabla fue adaptado a partir de template propuesto por (Banks, 2007)

El burndown del Sprint 4 (Figura 42) muestra que una vez fue finalizada la primera semana se habían cumplido 22 horas de trabajo, al finalizar la segunda semana se cumplieron 45 horas siguiendo un ritmo de trabajo que permitió completar el sprint al finalizar la tercera semana. Al igual que con el resto de sprints no se presentaron inconvenientes que retrasaran su ejecución.

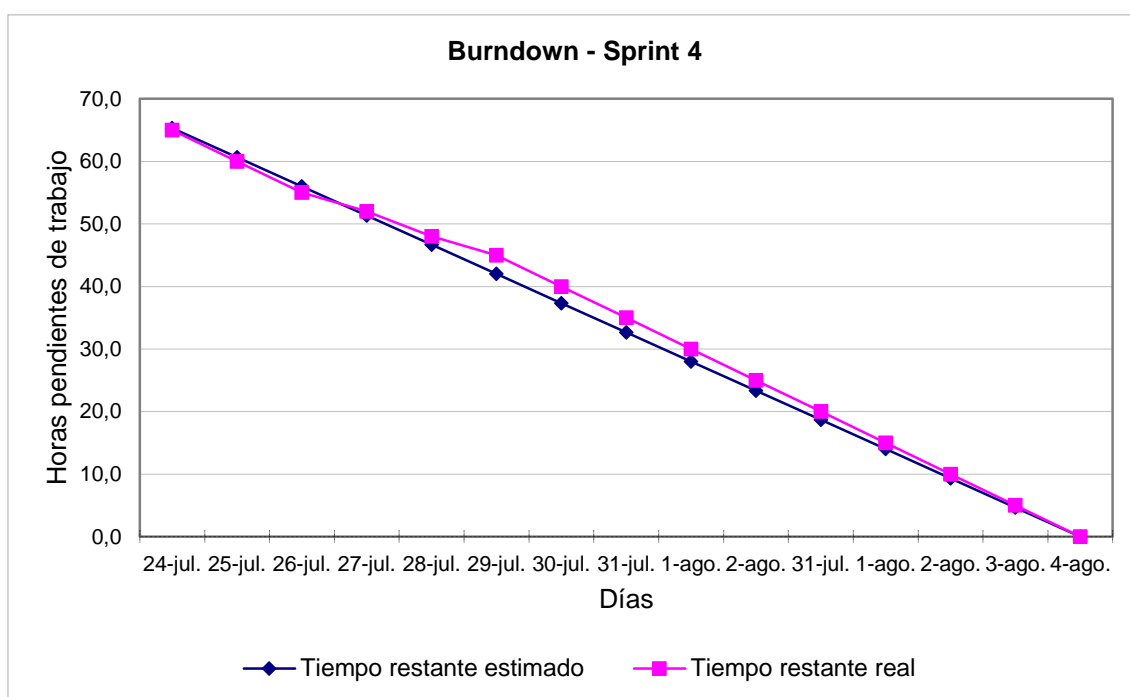


Figura 42. Burndown de Sprint 4

Fuente: Elaboración propia.

4.7. Plan de pruebas

Las tablas 32, 33, 34 y 35 especifican las pruebas realizadas, cada una incluye: *Sprint* indica el número de sprint, *Funcionalidad* es el código de la funcionalidad, *Id* es el identificador asignado al caso de prueba descrito, *Nombre* de la prueba, *Descripción* especifica el detalle de la prueba, *Prerrequisitos* indica los requisitos que deben haberse cumplido previamente para llevar a cabo la prueba, *Pasos* describe todas las actividades llevadas a cabo paso a paso para efectuar la prueba, *Resultado esperado* especifica el resultado esperado de la ejecución de la prueba y *Resultado obtenido* indica si la ejecución de la prueba fue o no satisfactoria.

Tabla 32. Plan de pruebas – Sprint 1

Sprint	Funcionalidad	Id	Nombre	Descripción	Prerrequisitos	Pasos	Resultado esperado	Resultado obtenido
1	F-1.1	CP01	Invocar método de API Rest que devuelve Misión de carrera	Invocar método a través de su URL, proveyendo código de carrera	- El método de API Rest que devuelve la misión de carrera fue creado	a) Ejecutar la aplicación Postman de Chrome b) Escribir la URL de la API especificando centroCosCodigo=00165	- Código de estado 200 - Cuerpo del mensaje devuelto incluye la misión de la CISIC	Prueba satisfactoria
1	F-1.1	CP02	Invocar método de API Rest que devuelve Visión de carrera	Invocar método a través de su URL, especificando el código de carrera	- El método de API Rest que devuelve la visión de carrera fue creado	a) Ejecutar la aplicación Postman de Chrome b) Escribir URL de la API especificando centroCosCodigo=00165	- Código de estado 200 - Cuerpo del mensaje devuelto incluye la visión de la CISIC	Prueba satisfactoria
1	F-1.1	CP03	Actualización de Misión y visión de carrera en el blog de la CISIC	Ejecutar función que actualiza misión y visión en el blog	- Se ha ejecutado con éxito CP01 y CP02 - La misión y/o visión publicadas en el blog son diferentes a las obtenidas del SIIU	a) Acceder al blog como un usuario con privilegios de edición b) Modificar la misión y/o visión publicadas de manera que difiera(n) de las almacenadas en el SIIU c) Ejecutar la función de actualización	- La misión y/o visión del blog son actualizadas con la información devuelta por las API's invocadas	Prueba satisfactoria
1	F-1.2	CP04	Invocar método de API Rest que devuelve Ficha de carrera	Invocar método a través de su URL, especificando los parámetros código de carrera y ciclo académico	- El método de API Rest que devuelve la ficha de carrera fue creado	a) Ejecutar la aplicación Postman de Chrome b) Escribir la URL de la API especificando codigo=00165 y ciclo=0317-0817	- Código de estado 200 - Cuerpo del mensaje devuelto contiene la ficha de la CISIC (nombre coordinador, correo, descripción)	Prueba satisfactoria

Sprint	Funcionalidad	Id	Nombre	Descripción	Prerrequisitos	Pasos	Resultado esperado	Resultado obtenido
1	F-1.2	CP05	Actualizar el contenido informativo Ficha de la carrera del blog de la CISIC	Ejecutar función que actualiza ficha de carrera en el blog	<ul style="list-style-type: none"> - Se ha ejecutado con éxito CP04 - La ficha de carrera publicada en el blog es diferente a la obtenida del SIIU 	<ul style="list-style-type: none"> a) Acceder al blog como un usuario con privilegios de edición b) Modificar Ficha de carrera de manera que difiera de la almacenada en el SIIU c) Ejecutar la función de actualización 	<ul style="list-style-type: none"> - La ficha de la CISIC (nombre coordinador, correo, denominación) es actualizada con la información devuelta por la API invocada 	Prueba satisfactoria

Fuente: Elaboración propia.

Tabla 33. Plan de pruebas – Sprint 2

Sprint	Funcionalidad	Id	Nombre	Descripción	Prerrequisitos	Pasos	Resultado esperado	Resultado obtenido
2	F-2.1	CP06	Invocar método de API Rest que devuelve Objetivos educacionales de la carrera	Invocar método a través de su URL, especificando código de carrera	- El método de API Rest que devuelve los objetivos educacionales fue creado	a) Ejecutar la aplicación Postman de Chrome b) Escribir URL/endpoint de la API especificando depenCodigo=00165	- Código de estado 200 - Cuerpo del mensaje devuelto contiene los objetivos educacionales de la CISIC	Prueba satisfactoria
2	F-2.1	CP07	Actualizar el contenido informativo Objetivos educacionales del blog de la CISIC	Ejecutar la función que actualiza objetivos educacionales en el blog	- Se ha ejecutado con éxito CP06 - Los objetivos educacionales publicados en el blog difieren de los obtenidos del SIU	a) Acceder al blog como un usuario con privilegios de edición b) Modificar los objetivos educacionales de manera que difieran de los almacenados en el SIU c) Ejecutar la función de actualización	- Los objetivos educacionales son actualizados con la información devuelta por la API invocada	Prueba satisfactoria
2	F-2.2	CP08	Invocar método de API Rest que devuelve resultados de aprendizaje de la carrera	Invocar método a través de su URL proporcionando código de carrera	- El método de la API que devuelve los resultados de aprendizaje fue creado	a) Ejecutar la aplicación Postman de Chrome b) Escribir URL/endpoint de la API especificando depenCodigo=00165	- Código de estado 200 - Cuerpo del mensaje devuelto contiene los resultados de aprendizaje de la CISIC	Prueba satisfactoria
2	F-2.2	CP09	Actualizar el contenido informativo Resultados de aprendizaje del blog de la CISIC	Ejecutar la función que actualiza resultados de aprendizaje en el blog	- Se ha ejecutado con éxito CP08 - Los Resultados de aprendizaje publicados en el blog son diferentes a los obtenidos desde el SIU	a) Acceder al blog como un usuario con privilegios de edición b) Modificar los resultados de aprendizaje de manera que difieran de los almacenados en el SIU c) Ejecutar la función de actualización	- Los resultados de aprendizaje son actualizados con la información devuelta por la API invocada	Prueba satisfactoria

Sprint	Funcionalidad	Id	Nombre	Descripción	Prerrequisitos	Pasos	Resultado esperado	Resultado obtenido
2	F-2.3	CP10	Invocar método de API Rest que devuelve malla curricular de la carrera	Invocar método a través de su URL, proveyendo código de carrera y ciclo académico	- El método de la API Rest que devuelve la malla curricular de carrera fue creado	a) Ejecutar la aplicación Postman de Chrome b) Escribir URL/endpoint de la API especificando dependencia=00165, ciclo=0317-0817	- Código de estado 200 - Cuerpo del mensaje devuelto contiene la malla curricular de la CISIC del ciclo académico 0317-0817	Prueba satisfactoria
2	F-2.3	CP11	Actualizar el contenido informativo Malla curricular del blog de la CISIC	Ejecutar la función que actualiza Malla curricular en el blog	- Se ha ejecutado con éxito CP10 - La Malla curricular publicada en el blog es diferente a la obtenida desde el SIU	a) Acceder al blog como un usuario con privilegios de edición b) Modificar la malla curricular de manera que difiera de la almacenada en el SIU c) Ejecutar la función de actualización	- La malla curricular es actualizada con la información devuelta por la API invocada	Prueba satisfactoria

Fuente: Elaboración propia.

Tabla 34. Plan de pruebas – Sprint 3

Sprint	Funcionalidad	Id	Nombre	Descripción	Prerrequisitos	Pasos	Resultado esperado	Resultado obtenido
3	F-3.1	CP12	Invocar método de API Rest que devuelve la carga horaria en investigación de los docentes de la carrera	Invocar método a través de su URL especificando código de carrera y ciclo académico	- El método de la API que devuelve la carga horaria de los docentes fue creado	a) Ejecutar la aplicación Postman de Chrome b) Escribir URL/endpoint de la API especificando código=00165 y ciclo=0317-0817	- Código de estado 200 - Cuerpo del mensaje devuelto contiene la carga horaria en investigación de los docentes de la CISIC	Prueba satisfactoria
3	F-3.1	CP13	Invocar método de la API Rest que devuelve la producción científica	Invocar el método a través de su URL proveyendo código de carrera y ciclo académico	- El método que devuelve la producción científica de los docentes de la carrera fue creado	a) Ejecutar la aplicación Postman de Chrome b) Escribir URL/endpoint de la API Producción científica, especificando código=00165, ciclo=0317-0817	- Código de estado 200 - Cuerpo del mensaje devuelto contiene la producción científica de los docentes de la CISIC durante el periodo 0317-0817	Prueba satisfactoria
3	F-3.1	CP14	Actualizar el contenido informativo Docentes investigadores / Producción científica del blog de la CISIC	Ejecutar la función que actualiza Docentes investigadores / Producción científica en el blog	- Se ha ejecutado con éxito CP12 y CP13 - Los Docentes investigadores / Producción científica publicados en el blog son diferentes a los obtenidos del SIIU	a) Acceder al blog como un usuario con privilegios de edición b) Modificar el contenido de docentes investigadores/ producción científica de manera que difiera del almacenado en el SIIU c) Ejecutar la función de actualización	- El contenido de docentes investigadores / producción científica es actualizado con la información devuelta por las API's invocadas	Prueba satisfactoria

Fuente: Elaboración propia.

Tabla 35. Plan de pruebas – Sprint 4

Sprint	Funcionalidad	Id	Nombre	Descripción	Prerrequisitos	Pasos	Resultado esperado	Resultado obtenido
4	F-4.1	CP15	Invocar método de API Rest que devuelve la carga horaria asociada a la vinculación de los docentes de la carrera	Invocar método a través de su URL especificando código de carrera y ciclo académico	- El método de la API que devuelve la carga horaria asociada a la vinculación de los docentes fue creado	a) Ejecutar la aplicación Postman de Chrome b) Escribir URL/endpoint de la API especificando dependencia=00165 y ciclo=0317-0817	- Código de estado 200 - Cuerpo del mensaje devuelto contiene la carga horaria en vinculación de los docentes de la CISIC	Prueba satisfactoria
4	F-4.1	CP16	Actualizar el contenido informativo Docentes en la vinculación del blog	Ejecutar la función que actualiza Docentes en la vinculación en el blog	- Se ha ejecutado con éxito CP15 - Los Docentes en la vinculación publicados en el blog son diferentes a los obtenidos del SIIU	a) Acceder al blog como un usuario con privilegios de edición b) Modificar el contenido de docentes en la vinculación de manera que difiera del almacenado en el SIIU c) Ejecutar la función de actualización	- El contenido de docentes en la vinculación es actualizado con la información devuelta por la API invocada	Prueba satisfactoria
4	F-4.2	CP17	Invocar método de API Rest que devuelve la información de movilidad docente de la carrera	Invocar la API Rest Movilidad docente a través de su URL proveyendo código de carrera	- El método de la API Rest que devuelve la información de movilidad docente fue creada	a) Ejecutar la aplicación Postman de Chrome b) Escribir la URL de la API especificando dependencia=00165	- Código de estado 200 - Cuerpo del mensaje devuelto contiene los datos de Movilidad docente de la CISIC	Prueba satisfactoria
4	F-4.2	CP18	Actualizar el contenido informativo Movilidad Docente del blog	Ejecutar la función que actualiza Docentes en la vinculación en el blog	- Se ha ejecutado con éxito CP17 - Los datos de movilidad docente publicados en el blog son diferentes a los obtenidos del SIIU	a) Acceder al blog como un usuario con privilegios de edición b) Modificar los datos de movilidad docente de manera que difieran de los almacenados en el SIIU c) Ejecutar la función de actualización	- Los datos de movilidad docente son actualizados con la información devuelta por la API invocada	Prueba satisfactoria

Sprint	Funcionalidad	Id	Nombre	Descripción	Prerrequisitos	Pasos	Resultado esperado	Resultado obtenido
4	F-4.3	CP19	Invocar método de API Rest que devuelve la información de convenios de la carrera	Invocar el método a través de su URL especificando código de carrera	- El método de la API Rest que devuelve los convenios de la carrera fue creado	a) Ejecutar la aplicación Postman de Chrome b) Escribir URL/endpoint de la API dependencia=00165	- Código de estado 200 - Cuerpo del mensaje devuelto contiene los datos de Convenios de la CISIC	Prueba satisfactoria
4	F-4.3	CP20	Actualizar el contenido informativo Convenios del blog	Ejecutar la función que actualiza convenios en el blog	- Se ha ejecutado con éxito CP19 - Los convenios publicados en el blog son diferentes a los obtenidos del SIU	a) Acceder al blog como un usuario con privilegios de edición b) Modificar los datos de convenios de manera que difieran de los almacenados en el SIU c) Ejecutar la función de actualización	- Los datos de convenios son actualizados con la información devuelta por la API invocada	Prueba satisfactoria

Fuente: Elaboración propia.

Las pruebas efectuadas permitieron comprobar el funcionamiento de los módulos implementados verificando que los resultados obtenidos correspondan con los planteados inicialmente.

4.8. Revisión de sprints

Las tablas 36, 37, 38 y 39 fueron actualizadas para los sprints conforme finalizaba la implementación de cada funcionalidad y completó todas sus tareas, los campos empleados se describen enseguida:

- **Sprint.**- Indica el número del sprint.
- **Id.**- Es el identificador de cada tarea.
- **Tarea.**- Contiene la descripción de las tareas definidas para el sprint.
- **Estado de tarea.**- Especifica si la tarea aún *No inicia*, está *En curso* o ya ha sido *Finalizada*.
- **Tarea Cumplida.**- Muestra si la tarea fue cumplida o no.
- **Resultado alcanzado por la tarea.**- Describe el resultado de la ejecución de la tarea implementada.
- **Funcionalidad implementada.**- Indica la funcionalidad alcanzada mediante la ejecución del conjunto de tareas relacionadas.
- **Resultado alcanzado por el sprint.**- Especifica el resultado alcanzado en el sprint, que a su vez corresponde a todas las funcionalidades logradas en el mismo.
- **Estado del Sprint.**- Indica si el sprint *No inicia* es decir, ninguna de sus tareas todavía se ha ejecutado, está *En curso* o ya ha *Finalizado* lo que implica que todas sus tareas se cumplieron satisfactoriamente.

Tabla 36. Revisión de resultados alcanzados - Sprint 1

Sprint	Id	Tarea	Estado de tarea	Tarea Cumplida		Resultado alcanzado por la tarea	Funcionalidad implementada	Resultado alcanzado por el sprint	Estado del Sprint
				Si	No				
1	1.1	Implementación API Rest Misión	Finalizada	✓		Al invocar el método del web service mediante su URL, la <i>misión</i> de la carrera es devuelta en formato JSON	Actualización del contenido informativo <i>Misión y visión</i> de la carrera perteneciente a la categoría	Los contenidos Ficha de carrera, y misión y visión se actualizan en el blog de la CISIC, cuando la información publicada no coincide con la almacenada en el SIIU	Finalizado
	1.2	Implementación API Rest Visión	Finalizada	✓		Al invocar el método del web service mediante su URL, la <i>visión</i> de la carrera es devuelta en formato JSON	Información del blog de la CISIC		
	1.3	Implementación funcionalidad para actualización del contenido Misión y Visión de Carrera del blog	Finalizada	✓		El método creado consume los métodos de los web services misión y visión, actualiza el contenido del blog cuando la información de éste y del SIIU no coincide			
	1.4	Implementación API Rest ConsultasSQL método que devuelve Ficha de carrera	Finalizada	✓		Al invocar el método del web service mediante su URL, la información de la <i>Ficha de carrera</i> es devuelta en formato JSON	Actualización del contenido informativo <i>Ficha de la carrera</i> perteneciente a la categoría Información del blog de la CISIC		
	1.5	Implementación funcionalidad para actualización del contenido Ficha de Carrera del blog	Finalizada	✓		El método creado consume el web service que devuelve la <i>ficha de carrera</i> , actualiza el contenido del blog cuando la información de éste y del SIIU no coinciden			

Fuente: Elaboración propia.

Tabla 37. Revisión de resultados alcanzados – Sprint 2

Sprint	Id	Tarea	Estado de tarea	Tarea Cumplida		Resultado alcanzado por la tarea	Funcionalidad implementada	Resultado alcanzado por el sprint	Estado del Sprint
				Si	No				
2	2.1	Implementación API Rest Objetivos Educativos	Finalizada	✓		Al invocar el método del web service mediante su URL, los <i>objetivos educativos</i> de la carrera son devueltos en formato JSON	Actualización del contenido informativo <i>Objetivos educativos</i>	Los contenidos Objetivos Educativos, Resultados de	Finalizado
	2.2	Implementación funcionalidad para actualización del contenido Objetivos Educativos del blog	Finalizada	✓		El método creado consume el web service que devuelve los <i>objetivos educativos</i> , actualiza el contenido correspondiente del blog cuando la información de éste y del SIIU no coincide	perteneciente a la categoría Gestión académica del blog de la CISIC	Aprendizaje y Malla Curricular se actualizan en el blog de la CISIC, cuando la información publicada no coincide con la almacenada en el SIIU	
	2.3	Implementación API Rest Resultados de aprendizaje	Finalizada	✓		Al invocar el método del web service mediante su URL, los <i>resultados de aprendizaje</i> de la carrera son devueltos en formato JSON	Actualización del contenido informativo <i>Resultados de aprendizaje</i>		
	2.4	Implementación funcionalidad para actualización del contenido Resultados de aprendizaje del blog	Finalizada	✓		El método creado consume el web service que devuelve los <i>Resultados de aprendizaje</i> , actualiza el contenido correspondiente del blog cuando la información de éste y del SIIU no coincide	perteneciente a la categoría Gestión académica del blog de la CISIC		
	2.5	Implementación API Rest ConsultasSQL método que devuelve Malla curricular	Finalizada	✓		Al invocar el método del web service mediante su URL, la <i>malla curricular</i> de la carrera es devuelta en formato JSON	Actualización del contenido informativo <i>Malla curricular</i>		
	2.6	Implementación funcionalidad para actualización del contenido Malla curricular	Finalizada	✓		El método creado consume el web service que devuelve la <i>malla curricular</i> , actualiza el contenido correspondiente del blog cuando la información de éste y del SIIU no coincide	perteneciente a la categoría Gestión académica del blog de la CISIC		

Fuente: Elaboración propia.

Tabla 38. Revisión de resultados alcanzados - Sprint 3

Sprint	Id	Tarea	Estado de tarea	Tarea Cumplida		Resultado alcanzado por la tarea	Funcionalidad implementada	Resultado alcanzado por el sprint	Estado del Sprint
				Si	No				
3	3.1	Implementación API Rest ConsultasSQL método que devuelve Carga horaria de docentes investigadores	Finalizada	✓		Al invocar el método del web service mediante su URL, la <i>Carga horaria de docentes</i> en investigación de la carrera son devueltos en formato JSON	Actualización del contenido informativo <i>Docentes investigadores/ Producción científica</i> perteneciente a la categoría	Los contenidos Docentes investigadores y Producción Científica se actualizan en el blog de la CISIC cuando la información publicada no coincide con la almacenada en el SIIU	Finalizado
	3.2	Implementación API Rest ConsultasSQL método que devuelve Producción científica de los docentes de la carrera	Finalizada	✓		Al invocar el método del web service mediante su URL, la <i>producción científica</i> de la carrera es devuelta en formato JSON	Investigación del blog de la CISIC		
	3.3	Implementación funcionalidad para actualización del contenido Docentes investigadores/Producción científica del blog	Finalizada	✓		El método creado consume los web services que devuelven los datos sobre <i>Docentes investigadores y producción científica</i> , actualiza el contenido del blog cuando la información de éste y del SIIU no coincide			

Fuente: Elaboración propia.

Tabla 39. Revisión de resultados alcanzados - Sprint 4

Sprint	Id	Tarea	Estado de tarea	Tarea Cumplida		Resultado alcanzado por la tarea	Funcionalidad implementada	Resultado alcanzado por el sprint	Estado del Sprint
				Si	No				
4	4.1	Implementación API Rest ConsultasSQL método que devuelve la carga horaria de Docentes en la vinculación	Finalizada	✓		Al invocar el método del web service mediante su URL, la información sobre horas de docentes destinadas a vinculación es devuelta como JSON	Actualización del contenido informativo <i>Docentes en la vinculación</i> perteneciente a la categoría Vinculación del blog de la CISIC	Los contenidos Docentes en la Vinculación, Movilidad Docente y Convenios se actualizan en el blog de la CISIC cuando la información publicada no coincide con la almacenada en el SIU	Finalizado
	4.2	Implementación funcionalidad para actualización del contenido Docentes en la vinculación del blog	Finalizada	✓		El método creado consume el web service que devuelve la <i>Carga horaria</i> , actualiza el contenido correspondiente del blog cuando la información de éste y del SIU no coincide			
	4.3	Implementación API Rest ConsultasSQL método que devuelve la información sobre movilidad docente de la carrera	Finalizada	✓		Al invocar el método del web service mediante su URL, la información sobre <i>movilidad docente</i> de la carrera es devuelta en formato JSON	Actualización del contenido informativo <i>Movilidad docente</i> perteneciente a la categoría Investigación del blog de la CISIC		
	4.5	Implementación funcionalidad para actualización del contenido Movilidad Docente del blog	Finalizada	✓		El método creado consume el web service que devuelve los datos de <i>movilidad docente</i> , actualiza el contenido correspondiente del blog cuando la información de éste y del SIU no coincide			
	4.4	Implementación API Rest ConsultasSQL método que devuelve Convenios de la carrera	Finalizada	✓		Al invocar el método del web service mediante su URL, los <i>convenios</i> de la carrera son devueltos en formato JSON	Actualización del contenido <i>Convenios</i> perteneciente a la categoría Investigación del blog de la CISIC		
	4.6	Implementación funcionalidad para actualización del contenido Convenios del blog	Finalizada	✓		El método creado consume el web service que devuelve los <i>Convenios</i> , actualiza el contenido correspondiente del blog cuando la información de éste y del SIU no coincide			

Fuente: Elaboración propia.

Las tablas anteriores (36 a 39) se emplearon para revisar los sprints después de que se cumplieron la totalidad de tareas definidas para cada uno, se compararon los resultados obtenidos versus los planteados inicialmente, cuando ambos coincidieron entonces el sprint pasó a estado finalizado.

4.9. Beneficios de la propuesta

El software implementado permite que el SIIU y blog de la CISIC compartan información que es empleado para actualizar varios contenidos informativos del blog. Después de que en la sección anterior se verificara el cumplimiento de los resultados esperados en cada sprint, a continuación se listan los beneficios brindados:

- **Actualización automática de contenidos.**- Se aprovecha la información disponible en los módulos del SIIU con el objeto de actualizar contenidos determinados en el blog de la CISIC de forma automática.
- **Información coincidente entre ambas aplicaciones.**- La actualización llevada a cabo garantiza que la información publicada en el blog coincida con la almacenada en la base de datos del SIIU.
- **Disminución de tiempos para actualizar contenidos.**- Debido a que la actualización de contenidos es automática, se logra disminuir el tiempo en el que se lleva a cabo dicha tarea; así mismo disminuye el periodo durante el cual pudieran no ser coincidentes los datos entre el blog y el SIIU.
- **Herramienta de apoyo a la gestión de contenidos.**- La información vinculada entre el blog y el SIIU no necesita ser actualizada directamente desde la herramienta de administración de contenidos del CMS WordPress.
- **Seguimiento de información vinculada.**- Se identificó la información que puede ser vinculada entre el SIIU y blog de la CISIC (ver Tabla 21), pero además quedaron descritos los demás contenidos para que conforme se habiliten otros módulos del SIIU que pudieran estar relacionados con esos contenidos se registre lo respectivo en la tabla diseñada.

4.10. Resultados del proyecto

La Tabla 40 presenta un resumen de las características de la propuesta de integración de aplicaciones implementada entre el SIIU y el blog de carrera del portal web, en la que se describe el tipo de aplicaciones integradas, nivel de integración efectuada, posibles modificaciones realizadas en las aplicaciones, entre otras; varias de las características incluidas podrían ser parte de criterios para valorar propuestas de integración de aplicaciones.

Tabla 40. *Características de la propuesta de integración implementada*

Características	Opciones	Descripción
Tipo de aplicaciones integradas	<input checked="" type="checkbox"/> Heterogéneas <input type="checkbox"/> Homogéneas	<p>Las aplicaciones integradas son heterogéneas, cumplen diferentes propósitos, las tecnologías con las que están desarrolladas difieren.</p> <ul style="list-style-type: none"> - Aplicación 1: Sistema Informático Integrado Universitario (Oracle Forms 11g, Oracle Reports 11g, Oracle Apex 4.2, Servidor de Aplicaciones Web Logic 11, Base de datos Oracle 11g) - Aplicación 2: Blog de carrera del portal web (Wordpress 4.7, servidor web Apache 2.2, base de datos MySQL 5.6.13)
Objetivo de la integración de aplicaciones	<input checked="" type="checkbox"/> Compartir datos <input type="checkbox"/> Compartir funcionalidades	<p>Objetivo de la integración</p> <ul style="list-style-type: none"> - Compartir datos entre el SIIU y blog de carrera del portal web para actualizar contenidos en el blog.
Nivel de integración	<input type="checkbox"/> Datos <input type="checkbox"/> Interfaz de usuario <input checked="" type="checkbox"/> Método <input checked="" type="checkbox"/> Aplicación	<p>Módulos implementados</p> <ul style="list-style-type: none"> - Módulo A: Implementación de APIs REST que exponen la información del SIIU. - Módulo B: Consumir las APIs REST implementadas, realizar procesamiento de información, hacer uso de API REST de WordPress para actualizar información.

Características	Opciones	Descripción
La integración de aplicaciones requirió realizar modificaciones en ellas	<input type="checkbox"/> Si <input checked="" type="checkbox"/> No	Las aplicaciones no sufrieron modificación alguna con el objeto de ser integradas.
Uso de protocolos y formatos de intercambio de datos comunes	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No	El protocolo y formato de datos comunes empleados son HTTP y JSON respectivamente. La API Rest de WordPress hace uso de JSON como formato de solicitud/respuesta. Las API Rest implementadas utilizan los verbos de HTTP.
Están implicadas diferentes bases de datos	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No	Las bases de datos implicadas son Oracle 11g y Mysql versión community bajo licencia GNU.

Fuente: Elaboración propia.

Una vez implementada la propuesta de integración entre el SIIU y blog de la CISIC, se aplicó una encuesta con la finalidad de recoger los criterios de los stakeholders (Web master, Administradores de los módulos SIIU académico, biblioteca, planificación estratégica institucional, oficina de relaciones internacionales, recursos humanos y vinculación, y Director del Departamento de Desarrollo Tecnológico e Informático) respecto a la misma.

Los resultados del instrumento aplicado (ver Anexo 2 – E) se presentan en la Tabla 41, el cálculo del total para cada pregunta comprendió la sumatoria del producto de la cantidad de respuestas obtenidas en cada opción por su valor asociado, el porcentaje se obtuvo en base a la relación del total con el máximo valor que podía alcanzarse en cada pregunta (40).

Tabla 41. Tabulación encuesta stakeholders

Pregunta	Factor (*)					Total	Porcentaje alcanzado
	1	2	3	4	5		
1. ¿Cuál es el nivel de importancia de la propuesta que integra el SIIU y blog de la CISIC del portal web, para actualizar contenidos del blog?	0	0	0	1	7	39,00	97,50
2. Valore el nivel de importancia de las características/beneficios de la actualización de contenidos en el blog de la CISIC mediante información proveniente del SIIU, según la escala presentada a continuación							
a) Disminución del tiempo en la actualización de contenidos	0	0	1	4	3	34,00	85,00
b) Información coincidente entre el blog de carrera y la base de datos del SIIU	0	0	0	1	7	39,00	97,50
c) No se requiere utilizar la herramienta de administración del CMS para actualizar los contenidos vinculados	0	0	0	3	5	37,00	92,50
d) Se aprovecha la información ya existente (del SIIU) para actualizar contenidos del blog	0	0	2	0	6	36,00	90,00
e) Los contenidos del blog que pueden ser actualizados desde el SIIU están plenamente identificados	0	0	0	2	6	38,00	95,00
f) Los módulos del SIIU que gestionan información que puede ser vinculada con el blog están identificados	0	0	1	2	5	36,00	90,00
g) Están identificadas las tablas que contienen la información vinculada entre las dos aplicaciones	0	0	1	4	3	34,00	85,00
h) Para efectuar la integración no fue necesario modificar las aplicaciones	0	0	4	2	2	30,00	75,00
i) Se diseñó un instrumento que describe la naturaleza de las categorías y contenidos del blog, y que incluye su contraparte del SIIU	0	0	1	2	5	36,00	90,00
j) Las tareas de integración quedan establecidas de manera que puedan modificarse y aplicarse a otros blogs de carrera	0	0	1	2	5	36,00	90,00
3. ¿Considera como desventaja la no utilización que prescindir de la propuesta implementada sería una desventaja para la actualización automática de contenidos en el blog de la CISIC?	0	0	0	2	6	38,00	95,00

Fuente: Elaboración propia.

En la pregunta 1 se consultó el nivel de importancia atribuido a la propuesta de integración implementada, los resultados mostrados en la Figura 43 indican que para la mayoría de consultados el caso de integración de aplicaciones es totalmente importante.

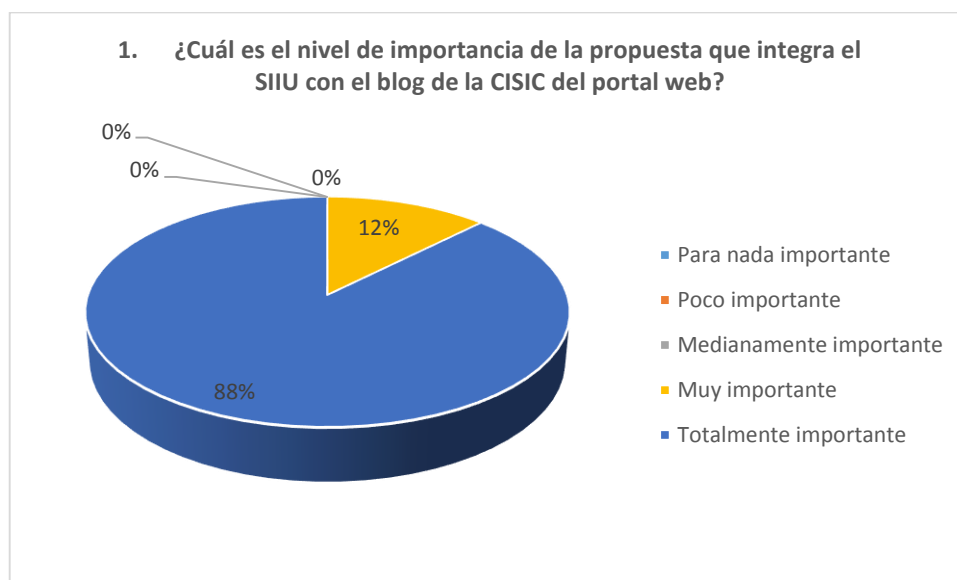
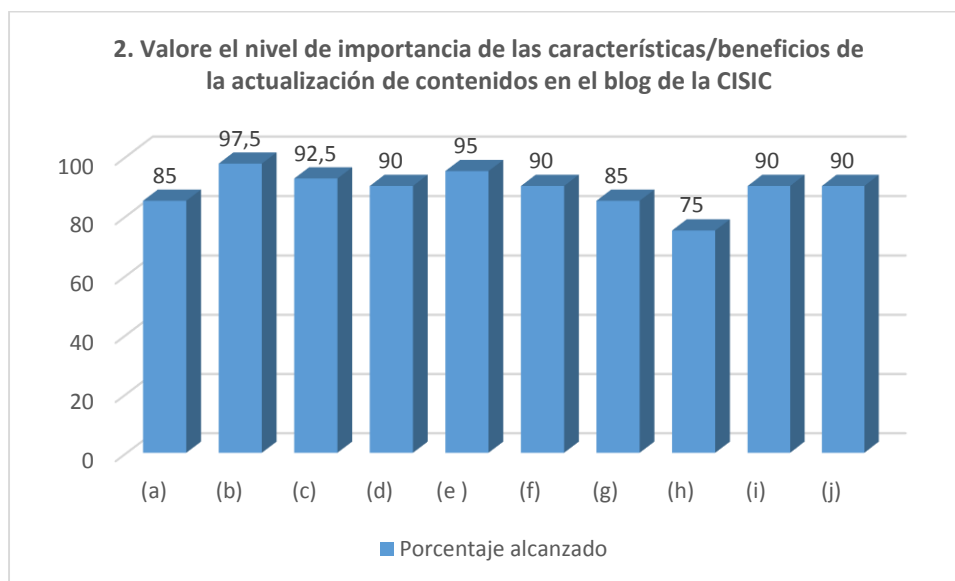


Figura 43. Tabulación – Pregunta 1

Fuente: Elaboración propia.

En cuanto a la importancia atribuida a las características/beneficios presentados en la pregunta 2, los resultados (Figura 44) indican que la opción (b) *Información coincidente entre el blog de carrera y la base de datos del SIIU* es considerada como la más importante (97,5%), seguida por (e) *Los contenidos del blog que pueden ser actualizados desde el SIIU están plenamente identificados* (95%) y (c) *No se requiere utilizar la herramienta de administración del CMS para actualizar los contenidos vinculados* (92.5%); mientras que la opción (h) *Para efectuar la integración no fue necesario modificar las aplicaciones* se le atribuye el porcentaje más bajo (75%).



- a) Disminución del tiempo en la actualización de contenidos
- b) Información coincidente entre el blog de carrera y la base de datos del SIIU
- c) No se requiere utilizar la herramienta de administración del CMS para actualizar los contenidos vinculados
- d) Se aprovecha la información ya existente (del SIIU) para actualizar contenidos del blog
- e) Los contenidos del blog que pueden ser actualizados desde el SIIU están plenamente identificados
- f) Los módulos del SIIU que gestionan información que puede ser vinculada con el blog están identificados
- g) Están identificadas las tablas que contienen la información vinculada entre las dos aplicaciones
- h) Para efectuar la integración no fue necesario modificar las aplicaciones
- i) Se diseñó un instrumento que describe la naturaleza de las categorías y contenidos del blog, y que incluye su contraparte del SIIU
- j) Las tareas de integración quedan establecidas de manera que puedan modificarse y aplicarse a otros blogs de carrera

Figura 44. Tabulación – Pregunta 2

Fuente: Elaboración propia.

En la pregunta 3, todos los encuestados coinciden en que no contar con la propuesta de integración implementada entre el SIIU y el blog de la CISIC del portal web sería una desventaja que incidiría sobre la actualización de contenidos.

Verificación de hipótesis

- **H₀:** La integración del SIIU y blog de la CISIC del portal web no incide en la actualización de contenidos en el blog de carrera.
- **H₁:** La integración del SIIU y blog de la CISIC del portal web incide en la actualización de contenidos en el blog de carrera.

En la tabla 42 se cruzaron preguntas correspondientes a cada variable con la finalidad de verificar o descartar la hipótesis nula.

Tabla 42. Tabla cruzada

Disminución del tiempo en la actualización de contenidos*2h. Para efectuar la integración no fue necesario modificar las aplicaciones						
		2h. Valore el nivel de importancia de: Para efectuar la integración no fue necesario modificar las aplicaciones			Total	
		Medianamente importante	Muy importante	Totalmente importante		
2a. Valore el nivel de importancia de: Disminución del tiempo en la actualización de contenidos	Medianamente importante	Recuento	0	0	1	1
		Recuento esperado	,5	,3	,3	1,0
		% del total	0,0%	0,0%	12,5%	12,5%
	Muy importante	Recuento	4	0	0	4
		Recuento esperado	2,0	1,0	1,0	4,0
		% del total	50,0%	0,0%	0,0%	50,0%
	Totalmente importante	Recuento	0	2	1	3
		Recuento esperado	1,5	,8	,8	3,0
		% del total	0,0%	25,0%	12,5%	37,5%
	Total	Recuento	4	2	2	8
		Recuento esperado	4,0	2,0	2,0	8,0
		% del total	50,0%	25,0%	25,0%	100,0%

Fuente: SPSS, en base a datos de la Tabla 41.

Tabla 43. Pruebas de chi-cuadrado

	Valor	df	Significación asintótica (bilateral)
Chi-cuadrado de Pearson	10,667 ^a	4	,031
Razón de verosimilitud	12,816	4	,012
Asociación lineal por lineal	,091	1	,763
N de casos válidos	8		

a. 9 casillas (100,0%) han esperado un recuento menor que 5. El recuento mínimo esperado es ,25.

Fuente: SPSS, en base a Tabla 42.

Con los datos obtenidos se rechaza la hipótesis nula debido a que la significación (ver Tabla 43) cumple la condición de ser menor o igual a 0,05; por lo tanto, se acepta la hipótesis alternativa misma que indica que las variables tienen relación.

La propuesta desarrollada cumplió con los objetivos tanto general como específicos detallados inicialmente, se configuraron los ambientes de desarrollo, se implementaron API's Rest para exponer información del SIIU y con dichos datos se actualizó contenidos en el blog de la Carrera de Ingeniería en Sistemas Computacionales de la UTN.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

La investigación desarrollada se centró en la integración de aplicaciones, tal como se mencionó en el capítulo II ésta puede convertirse en una necesidad para instituciones que cuentan con aplicaciones heterogéneas e información que regularmente se encuentra almacenada en diferentes bases de datos. Las conclusiones alcanzadas se explican enseguida:

- La propuesta de integración desarrollada entre el Sistema Informático Integrado Universitario (base de datos Oracle) y el blog de la Carrera de Ingeniería en Sistemas Computacionales (Wordpress) del portal web permite que ambas aplicaciones compartan información con la finalidad de actualizar contenidos en el blog; por tanto es una herramienta que apoya la gestión de contenidos y permite que la información publicada en el blog se encuentre a la par de la almacenada en el SIIU.
- Los resultados de la encuesta aplicada (Anexo 2 –D) indican que el blog de la CISIC es frecuentemente visitado, se lo considera importante y cumple con el objetivo de informar sobre la carrera y sus actividades; pero cerca de la mitad de los consultados (45%) manifestaron desconocer si la información publicada se encontraba actualizada.
- Los elementos del blog de la CISIC que pueden ser actualizados desde el SIIU son ficha, misión/visión de la carrera, objetivos educacionales, resultados de aprendizaje, malla curricular, docentes investigadores/producción científica, docentes en la vinculación, movilidad docente y convenios.
- Los servicios web permiten que varias aplicaciones se integren para compartir información sin importar plataformas ni tecnologías; sin embargo es necesario el uso de protocolos y formatos de intercambio de datos comunes, que para el caso particular de la propuesta implementada fueron HTTP y JSON; el software

implementado a través de API Rest expone información del SIIU que es consumida, procesada y llevada a la base de datos del blog a través del API Rest de WordPress.

- Las API Rest constituyen una alternativa menos compleja frente a los servicios web basados en SOAP, es la opción cuando se requieren formatos para intercambio de datos diferentes a XML; se caracteriza por la separación del cliente y servidor de manera que se puedan desarrollar de forma independiente además existe un bajo consumo de recursos.
- Node.js cuenta con una buena cantidad de módulos que apoyan el trabajo de los desarrolladores, tales como request mediante el cual se realizaron peticiones a endpoints específicos tanto de las API's Rest implementadas como la de WordPress; a través del módulo wpapi se efectuaron operaciones de edición en posts publicados en el blog utilizando el método update.
- La API Rest de WordPress es una poderosa herramienta que facilita la integración del CMS con diferentes tipos de aplicaciones; cuando se encuentra activada es posible interactuar con ella desde cualquier lenguaje, ya no limitándose exclusivamente a PHP, basta que el lenguaje cuente con soporte para operaciones a través del protocolo HTTP y soporte a JSON.
- Las metodologías ágiles de desarrollo no están exentas de tareas de planificación y seguimiento; en el caso específico de SCRUM los eventos y artefactos definidos ayudan a gestionar la planificación, realizar el seguimiento del avance de los sprints, verificar el cumplimiento de los resultados definidos inicialmente y revisar cada sprint con el objetivo de aplicar las lecciones aprendidas en sprints siguientes y futuros proyectos.

5.2. Recomendaciones

- El DDTI puede tomar como base la propuesta de integración efectuada entre el SIIU y blog de la CISIC, realizar los ajustes necesarios para aplicarlos con otros blogs de carreras; debido a que el 83% de los blogs que componen el portal web en la UTN sigue una estructura similar en cuanto a organización y contenidos publicados al del caso de estudio.

- En el DDTI los instrumentos diseñados para la revisión de blogs de carrera y sus contenidos (Tabla 21 y Anexo 1) deberán ser actualizados conforme se modifique la organización y contenidos de blogs que no siguen la estructura estándar (Administración, Contabilidad, Economía, Mercadotecnia, Gastronomía y Turismo) o se agreguen nuevos blogs; o cuando se desarrollen nuevos módulos del SIIU que gestionen datos que puedan emplearse para actualizar contenidos de los blogs.
- En relación a la integración de aplicación, WordPress y API Rest son varios temas que podrían ser considerados para investigaciones a futuro:
 - Extender/personalizar la API Rest de WordPress para tareas específicas.
 - Integrar WordPress con aplicaciones móviles o web, de manera que sea posible alimentar los blogs de carrera desde ese tipo de aplicaciones.
 - Integración de blogs de carrera para que compartan información en común, esto podría efectuarse entre blogs pertenecientes a una misma facultad.
 - Construcción de aplicaciones personalizadas encaminadas a la gestión de contenidos en WordPress (sin utilizar la herramienta de administración).

BIBLIOGRAFÍA

- Banks, R. (2007). *Odd-e Pte. Ltd.* Obtenido de Backlog templates: https://www.odd-e.com/resources/template_backlogs/rb_product_backlog_template.xls
- Baskaran, B. (15 de Septiembre de 2014). Added Value Framework and Information Technology Capabilities of Enterprise Application Integration. *A Study on ERP, CRM and Multi-Channel Commerce application integration in Fast Moving Consumer Goods (FMCG) Industry*. Delft, Países Bajos. Obtenido de <https://repository.tudelft.nl/islandora/object/uuid:f37ee12e-17d4-4b81-b89e-0dbdc22f3eb8/datastream/OBJ/download>
- California Department of Technology. (2 de Enero de 2014). *Enterprise Application Integration (EAI) - Reference Architecture (RA)*. Obtenido de California Department of Technology: <http://www.cio.ca.gov/ea/docs/Enterprise-Applications-Integration-EAI-V1.pdf>
- CEAACES. (2013). *CEAACES*. Obtenido de CEAACES: <http://www.ceaaces.gob.ec/sitio/wp-content/uploads/2013/10/Fichas-Te%CC%81cnicas.pdf>
- Chalam, M. (11 de Diciembre de 2012). *Explore The SharePoint*. Obtenido de Basics of Web Services: <http://sharepoint-works.blogspot.com/2012/12/basics-of-web-services.html>
- Chrzanowska, N. (17 de Marzo de 2017). *Netguru*. Obtenido de 10 Top Companies that Used Node.js in Production (examples): <https://www.netguru.co/blog/top-companies-used-nodejs-production>
- Departamento de Ciencia de la Computación e Inteligencia Artificial - Universidad de Alicante. (26 de Junio de 2014). *Universidad de Alicante*. Obtenido de Título de Experto Universitario en Desarrollo de Aplicaciones y Servicios con Java EE: <http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/sesion01-apuntes.pdf>
- Derecho Ecuador. (Agosto de 2008). *Revista Judicial Derecho Ecuador*. Obtenido de <http://www.derechoecuador.com/productos/producto/catalogo/registros-oficiales/2008/julio/code/18939/registro-oficial-no-393---jueves-31-de-julio-de-2008-primer-suplemento>
- Dirección de Informática. (2013). *Plan Integral de Desarrollo Informático 2013-2017*. Ibarra, Imbabura, Ecuador.
- Dirección General de Tecnologías de la Información - Generalitat Valenciana. (Octubre de 2016). *Generalitat Valenciana*. Obtenido de Sistema Integral Multicanal de Atención al Ciudadano. e-SIRCA-Desarrollo y consumo de servicios web.: http://www.dgtic.gva.es/documents/162018290/162077419/Desarrollo_y_consumo_de_servicios_web_Buenas_pr%C3%A1cticas_v16.pdf/b1962b97-00be-4b6f-82cf-58bda3b6b804

- Du, Y., Peng, W., & Zhou, L. (2008). Enterprise Application Integration: An Overview. *International Symposium on Intelligent Information Technology Application Workshops*, 1-14.
- Fredrich, T. (2017). *What is REST?* Obtenido de Restapitutorial.com: <http://www.restapitutorial.com/lessons/whatisrest.html#>
- Fu, G. (2016). FCA based ontology development for data integration. *Information Processing and Management*, 765-782.
- Fuseau, A. (2015). *APLICACIÓN DE UNA ARQUITECTURA ORIENTADA A SERVICIOS PARA LA INTEROPERABILIDAD DE LOS SERVICIOS DE SALUD PÚBLICOS Y PRIVADOS DEL ECUADOR*. Obtenido de Universidad de las Américas: <http://dspace.udla.edu.ec/bitstream/33000/3383/1/UDLA-EC-TMGSTI-2015-14%28S%29.pdf>
- IBM Knowledge Center. (2017). *Relación entre UDDI y WSDL*. Obtenido de IBM Knowledge Center: https://www.ibm.com/support/knowledgecenter/es/SSRTLW_9.1.0/org.eclipse.jst.ws.consumption.ui.doc.user/concepts/cwsdlud.html
- IETF. (Junio de 2014). *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*. Obtenido de Internet Engineering Task Force (IETF): <https://tools.ietf.org/pdf/rfc7231.pdf>
- Jovanovic, J. (2015). *Toptal*. Obtenido de Let LoopBack Do It: A Walkthrough of the Node API Framework You've Been Dreaming Of: <https://www.toptal.com/nodejs/let-loopback-do-it-a-walkthrough-of-the-node-api-framework-you-ve-been-dreaming-of>
- Kaviya, B., & Selvakumar, G. (2015). A Survey on RESTful Web Services Composition. *International Journal of Advanced Research in Computer Science and Software Engineering*, 98-100.
- Kumar, D. (2015). Best Practices for Building RESTful Web services. Bangalore, India: Infosys Limited. Obtenido de <https://www.infosys.com/digital/insights/Documents/restful-web-services.pdf>: <https://www.infosys.com/digital/insights/Documents/restful-web-services.pdf>
- Lin, F. (2005). Enterprise Application Integration (EAI) Techniques. Londres, Inglaterra: UCL Department of Computer Science. Obtenido de <http://www0.cs.ucl.ac.uk/staff/ucacwxe/lectures/3C05-04-05/EAI-Essay.pdf>
- Linthicum, D. (2003). *Next Generation Application Integration: From Simple Information to Web Services*. Addison Wesley.
- LoopBack . (Octubre de 2017). *LoopBack 2.x | LoopBack Documentation*. Obtenido de <http://loopback.io/doc/en/lb2/index.html>

- Menzinsky, A., López, G., & Palacio, J. (Julio de 2016). *Scrum Manager. Guía de formación*. Obtenido de [scrummanager.net](http://www.scrummanager.net):
http://www.scrummanager.net/files/sm_proyecto.pdf
- Moturi, C., Githehu, G., & Kahonge, A. (2013). Process Model for Enterprise Application Integration: Case for a Customs Department. *International Journal of Applied Information Systems (IJ AIS)*, 1-16.
- Mudrakola, S. (21 de Febrero de 2017). *techgenix.com*. Obtenido de Manage your data with these 3 open source ETL tools: <http://techgenix.com/open-source-etl-tools/>
- MuleSoft. (2014). *Applications Integration*. Obtenido de MuleSoft:
<https://www.mulesoft.com/resources/esb/applications-integration>
- Node.js Foundation. (2017). *Node.js*. Obtenido de <https://nodejs.org/en/>
- OASIS. (12 de Octubre de 2010). *Reference Model for Service Oriented Architecture 1.0*. Obtenido de OASIS: <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>
- Oracle. (Julio de 2009). *Oracle Database Heterogeneous Connectivity User's Guide 11g Release 2 (11.2)*. Obtenido de Oracle Corporation:
https://docs.oracle.com/cd/E11882_01/server.112/e11050.pdf
- Oracle. (Junio de 2014). *A Brave New Integration World. A Unified Framework for Cloud, Mobile, Internet of Things, Application Integration, and B2B*. Obtenido de Oracle Corporation: <http://www.oracle.com/us/products/middleware/soa/soa-simplify-enterprise-integration-2209678.pdf>
- Patil, N., Kshirsagar, M., & Jaypal, P. (2012). Enterprise Application Integration using Service Oriented Architecture with Generic Pattern. *International Journal of Current Engineering and Technology*, 97-103.
- Poulson, I. (18 de Abril de 2016). *Delicious Brains*. Obtenido de Tour of the WordPress Database: <https://deliciousbrains.com/tour-wordpress-database/>
- PressTigers. (2017). *Insights about WordPress 4.7 REST API Endpoints*. Obtenido de PressTigers: <http://www.prestigers.com/insights-about-wordpress-4-7-rest-api-endpoints/>
- REST API Handbook. (2017). *REST API Handbook*. Obtenido de WordPress Developer Resources: <https://developer.wordpress.org/rest-api/>
- RestPatterns. (24 de Abril de 2017). *HTTP Status Codes*. Obtenido de RestPatterns: http://www.restpatterns.org/HTTP_Status_Codes
- RisingStack. (2017). *RisingStack: Node.js Consulting, Development*. Obtenido de Resources Node_Hero: https://resources.risingstack.com/Node_Hero+-+The+complete+Node.js+tutorial+series+from+RisingStack.pdf
- Rosa Sequeira, F., Frantz, R., Yevseyeva, I., Emmerichd, M., & Basto-Fernandes, V. (2015). An EAI based integration solution for science and research outcomes information management. *Procedia Computer Science*, 894-901.

- Schwaber, k., & Sutherland, J. (Julio de 2013). *La Guía Definitiva de Scrum: Las Reglas del Juego*. Obtenido de [scrumguides.org](http://www.scrumguides.org):
<http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-es.pdf>
- SERVICIOS WEB ALICANTE - dwalicante. (2016). *Dwalicante.es*. Obtenido de <http://www.dwalicante.es/noticias-dise%C3%B1o-web/servicios-web-alicante/95>
- Solutions Review. (2017). *Solutions Review Buyers Guides and Best Practices*. Obtenido de Data Integration. Buyer's Guide.:
http://solutionsreview.com//dl/2017_Solutions_Review_Data_Integration_Buyers_Guide_SC66.pdf
- Sommerville, I. (2011). *INGENIERÍA DE SOFTWARE*. México: PEARSON EDUCACIÓN.
- SPAN Systems Corporation. (2014). *Enterprise Application Integration*. Obtenido de TutorialsPoint: <https://www.tutorialspoint.com/assets/white-papers/459.pdf>
- Srinivasan , K. (26 de Abril de 2014). *JAVABEAT*. Obtenido de WebServices : REST vs SOAP: <http://javabeat.net/webservices-rest-vs-soap/>
- StrongLoop. (2017). *The Node.js API Framework*. Obtenido de <https://loopback.io/>
- Tutorials Point. (2015). *Node.js Tutorial*. Obtenido de Tutorials Point (I) Pvt. Ltd: https://www.tutorialspoint.com/nodejs/nodejs_tutorial.pdf
- Tutorials Point. (2015). *Web Services Tutorial*. Obtenido de Tutorials Point (I) Pvt. Ltd.: <https://www.tutorialspoint.com/webservices/index.htm>
- Universidad Técnica del Norte. (2016). *UTN en cifras*. Recuperado el 21 de Febrero de 2016, de <http://svrapp3.utn.edu.ec:7001/apex/f?p=224:3:0::NO::>
- W3C. (s.f.). *Guía Breve de Servicios Web*. Obtenido de W3C España: <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>
- W3C Working Draft. (14 de Diciembre de 2002). *Web Services Architecture*. Obtenido de World Wide Web Consortium: <https://www.w3.org/TR/2002/WD-ws-arch-20021114/>
- WordPress Codex. (2017). *Database Description*. Obtenido de WordPress Codex: https://codex.wordpress.org/Database_Description
- WordPress Codex. (2017). *WordPress Features*. Obtenido de WordPress Codex: https://codex.wordpress.org/WordPress_Features

Anexo 2 – Cuestionarios de encuestas y entrevistas

A.- Entrevista dirigida al web máster

1. ¿Cuáles son los servicios que brinda el portal web de la UTN?
2. ¿Cuántos y cuáles son los blogs de carreras a los que se tiene acceso desde el portal web institucional?
3. ¿Cuántas personas están dedicadas a la administración del portal web (sitios, micro-sitios) institucional? ¿Qué tareas específicas realizan?
4. ¿En relación a los blogs de carreras que oferta la universidad, qué tipos de contenidos se publican en ellos?
5. ¿Cuáles son las herramientas CMS y base de datos con las que están desarrollados los blogs de carreras?
6. ¿Considera beneficiosa la implementación de un mecanismo (software) que permita la actualización de contenidos informativos en el blog de la CISIC a partir de información disponible en el SIIU? Si/No ¿Por qué?

B.- Entrevista dirigida al Director de Desarrollo Tecnológico e Informático

1. ¿Cuáles son las herramientas con las que se encuentra desarrollado el SIIU? (servidor_app, BD - software-versiones-ediciones)
2. ¿Cuántas personas del Departamento de TI están dedicadas a la administración del portal web institucional?
3. ¿En el SIIU existe información que podría aprovecharse para alimentar contenidos de los blogs de carreras? ¿Cuál es?
4. ¿Cuántos módulos componen el SIIU? ¿Cuáles son?
5. ¿De los módulos especificados en la pregunta anterior cuáles gestionan información que podría utilizarse para actualizar los blogs de carreras? ¿Qué información manejan?
6. ¿Cuántos técnicos de TI administran los módulos correspondientes a la pregunta 6?
7. ¿Considera beneficiosa la implementación de un mecanismo (software) que permita la actualización de contenidos informativos en el blog de la CISIC a partir de información disponible en el SIIU? Si/No ¿Por qué?

C.- Entrevista dirigida a los administradores de módulos del SIU

1. Indique el módulo del SIU que usted administra:

- ACA ORI RHU
 BIB PEI VIN

2. ¿Cuál es el objetivo del módulo que usted administra?

.....

.....

.....

3. Considerando la información publicada en el blog de la Carrera de Ingeniería en Sistemas Computacionales, del siguiente listado seleccione el (los) ítem(s) que podrían ser obtenidos del módulo que usted administra:

Categoría	Opción	Descripción
Información	Nuestra Carrera	<input type="checkbox"/> Fotografía coordinador <input type="checkbox"/> Mensaje coordinador <input type="checkbox"/> Hoja de vida coordinador en formato (PDF)
	Ficha de Carrera	<input type="checkbox"/> Denominación <input type="checkbox"/> Modalidad <input type="checkbox"/> Duración <input type="checkbox"/> Resolución de aprobación <input type="checkbox"/> Título que otorga <input type="checkbox"/> Coordinador de carrera <input type="checkbox"/> Email coordinador <input type="checkbox"/> Email carrera <input type="checkbox"/> Teléfono carrera
	Historia de la Carrera	<input type="checkbox"/> Fotografía docentes <input type="checkbox"/> Historia <input type="checkbox"/> Proyecto de creación de la carrera
	Misión y visión	<input type="checkbox"/> Misión de la carrera <input type="checkbox"/> Visión de la carrera
	Estructura Organizacional	<input type="checkbox"/> Organigrama de la carrera
	Plan estratégico de carrera	<input type="checkbox"/> Plan estratégico de la carrera: Datos informativos, misión, visión, políticas, mallas curriculares, distributivo, calendario, perfil de egreso, resultados y logros, líneas de investigación, proyectos de investigación, capacitaciones, programas y proyectos de vinculación (PDF).
	Laboratorios/Infraestructura	<input type="checkbox"/> Informe sobre laboratorios: Descripción laboratorios de informática de la facultad, cantidad de computadoras, organigrama, responsables, horarios de atención, funciones, áreas físicas, software disponible, fotografías (PDF).
Gestión académica	Perfil de Ingreso	<input type="checkbox"/> Perfil de ingreso
	Perfil de Egreso	<input type="checkbox"/> Perfil de egreso
	Perfil Profesional	<input type="checkbox"/> Perfil profesional
	Redes Académicas	<input type="checkbox"/> Redes académicas
	Campo Ocupacional	<input type="checkbox"/> Campo ocupacional
	Objetivos Educativos	<input type="checkbox"/> Objetivos educativos
	Resultados de Aprendizaje	<input type="checkbox"/> Resultados de aprendizaje

	Diseño/Rediseño Curricular	<input type="checkbox"/> Informe de diseño y rediseño curricular (PDF)
	Comisión Asesora	<input type="checkbox"/> Integrantes comisión asesora
	Malla Curricular	<input type="checkbox"/> Malla curricular de la carrera (PDF)
Investigación	Plan de Investigación	<input type="checkbox"/> Objetivos, líneas, sub-líneas
	Políticas de Investigación	<input type="checkbox"/> Políticas de investigación de la carrera
	Líneas de Investigación	<input type="checkbox"/> Líneas de investigación de la carrera
	Programas de Investigación	<input type="checkbox"/> Programas de investigación de la carrera
	Proyectos de Investigación	<input type="checkbox"/> Proyectos de investigación realizados por docentes (Nombre del Proyecto, línea de investigación, carreras involucradas, fecha inicio, fecha fin, estado, alcance territorial, director del proyecto, equipo de trabajo)
	Docentes investigadores/Producción Científica	<input type="checkbox"/> Nombres de docentes con carga horaria en investigación
	Trabajos de Titulación	<input type="checkbox"/> Nombres de los trabajos de titulación de la carrera
Vinculación	Plan de vinculación	<input type="checkbox"/> Documento PDF que contiene el plan de investigación de la carrera
	Docentes en la Vinculación	<input type="checkbox"/> Nombres de docentes con carga horaria en vinculación
	Movilidad Docente	<input type="checkbox"/> Fotografía y texto
	Movilidad Estudiantil	<input type="checkbox"/> Fotografía y texto
	Convenios	<input type="checkbox"/> Convenios
	Prácticas Pre-profesionales	<input type="checkbox"/> Acciones a ejecutar para la realización y aprobación de las Prácticas Pre profesionales <input type="checkbox"/> Requisitos para presentación del informe final de prácticas pre profesionales
	Giras Académicas	<input type="checkbox"/> Informe de giras académicas realizadas por la carrera (PDF)
	Graduados CISIC	<input type="checkbox"/> Listado de graduados de la carrera

4. **¿Cuáles son los scripts que permiten obtener la información señalada en el ítem anterior?**
5. **¿Considera beneficiosa la implementación de un mecanismo (software) que permita la actualización de contenidos informativos en el blog de la CISIC a partir de información disponible en el SIU? Si/No ¿Por qué?**

D.- Encuesta aplicada a docentes y estudiantes

1. **Seleccione la frecuencia con la que accede al blog de la CISIC.**
 - Muy frecuentemente
 - Frecuentemente
 - Ocasionalmente
 - Raramente
 - Nunca

2. **¿Considera que el blog de la CISIC cumple con el objetivo de informar sobre la carrera y sus actividades?**
 - Si
 - No
 - No sabe

3. **¿Cuál es el nivel de importancia que usted atribuye al blog para la Carrera?**
 - Muy importante
 - Importante
 - Medianamente importante
 - Poco importante
 - Nada importante

4. **¿Considera que la información publicada en el blog de la CISIC se encuentra actualizada?**
 - Si
 - No
 - No sabe

5. **De la categoría Información seleccione los ítems que usted consulta con mayor frecuencia en el blog de la CISIC.**
 - Misión/Visión de la carrera
 - Ficha de la carrera (Denominación, Coordinador, Correo electrónico)
 - Estructura organizacional
 - Plan de la carrera
 - Carrera en cifras

6. **De la categoría GESTIÓN ACADÉMICA seleccione los ítems que usted consulta con mayor frecuencia en el blog de la CISIC.**
 - Perfil de ingreso/egreso/profesional
 - Objetivos educacionales
 - Resultados de aprendizaje
 - Malla curricular
 - Plan de estudios/Sílabos

7. **De la categoría INVESTIGACIÓN seleccione los ítems que usted consulta con mayor frecuencia en el blog de la CISIC.**
 - Planes/Políticas/Líneas/Programas/Redes de Investigación
 - Proyectos de Investigación
 - Docentes investigadores/Producción Científica
 - Trabajos de Titulación

8. **De la categoría VINCULACIÓN seleccione los ítems que usted consulta con mayor frecuencia en el blog de la CISIC.**
 - Plan de vinculación
 - Docentes en la Vinculación
 - Movilidad Docente/Estudiantil

- Convenios
- Prácticas Pre-profesionales
- Giras Académicas
- Graduados CISIC

E.- Encuesta aplicada a Stakeholders

1. ¿Cuál es el nivel de importancia de la propuesta que integra el SIIU y blog de la CISIC del portal web, para actualizar contenidos del blog?

Totalmente importante (5)	Muy importante (4)	Medianamente importante (3)	Poco importante (2)	Para nada importante (1)
------------------------------	-----------------------	--------------------------------	------------------------	-----------------------------

2. Valore el nivel de importancia de las características/beneficios de la actualización de contenidos en el blog de la CISIC mediante información proveniente del SIIU, según la escala presentada a continuación:

Totalmente importante (5)	Muy importante (4)	Medianamente importante (3)	Poco importante (2)	Para nada importante (1)
------------------------------	-----------------------	--------------------------------	------------------------	-----------------------------

Opciones	(5)	(4)	(3)	(2)	(1)
a) Disminución del tiempo en la actualización de contenidos					
b) Información coincidente entre el blog de carrera y la base de datos del SIIU					
c) No se requiere utilizar la herramienta de administración del CMS para actualizar los contenidos vinculados					
d) Se aprovecha la información ya existente (del SIIU) para actualizar contenidos del blog					
e) Los contenidos del blog que pueden ser actualizados desde el SIIU están plenamente identificados					
f) Los módulos del SIIU que gestionan información que puede ser vinculada con el blog están identificados					
g) Están identificadas las tablas que contienen la información vinculada entre las dos aplicaciones					
h) Para efectuar la integración no fue necesario modificar las aplicaciones					
i) Se diseñó un instrumento que describe la naturaleza de las categorías y contenidos del blog, y que incluye su contraparte del SIIU					
j) El proceso de integración podría aplicarse a otras					

3. ¿Considera que prescindir de la propuesta implementada sería una desventaja para la actualización automática de contenidos en el blog de la CISIC?

Totalmente de acuerdo (5)	Muy de acuerdo (4)	Medianamente de acuerdo (3)	Poco de acuerdo (2)	Para nada de acuerdo (1)
------------------------------	-----------------------	--------------------------------	------------------------	-----------------------------