



UNIVERSIDAD DE ORIENTE
NUCLEO NUEVA ESPARTA
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
LICENCIATURA EN INFORMÁTICA

DESARROLLO DE UNA WEB API (INTERFAZ DE PROGRAMACIÓN DE APLICACIONES) CON ENFOQUE REST (TRANSFERENCIA DE ESTADO REPRESENTACIONAL) PARA LA GESTIÓN DE INFORMACIÓN ACADÉMICA ESTUDIANTIL, COMO COMPLEMENTO AL SISTEMA ACADÉMICO EN LÍNEA (SAEL) DE LA UNIVERSIDAD DE ORIENTE – NÚCLEO DE NUEVA ESPARTA.

Trabajo de Grado Modalidad Tesis de Grado presentado como requisito parcial para optar al Título de LICENCIADO EN INFORMÁTICA.

Br. Roger Aldayl Patiño Blanco

Guatamare, Julio de 2017.



UNIVERSIDAD DE ORIENTE
NÚCLEO NUEVA ESPARTA
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
LICENCIATURA EN INFORMÁTICA

ACTA DE APROBACIÓN

Quienes suscriben, miembros del jurado evaluador designado para examinar el Trabajo de Grado Modalidad Tesis de Grado, titulado: DESARROLLO DE UNA WEB API (INTERFAZ DE PROGRAMACIÓN DE APLICACIONES) CON ENFOQUE REST (TRANSFERENCIA DE ESTADO REPRESENTACIONAL) PARA LA GESTIÓN DE INFORMACIÓN ACADÉMICA ESTUDIANTIL, COMO COMPLEMENTO AL SISTEMA ACADÉMICO EN LÍNEA (SAEL) DE LA UNIVERSIDAD DE ORIENTE – NÚCLEO DE NUEVA ESPARTA, presentado por el Br. Roger Aldayl Patiño Blanco (C.I.: 21.015.369). Una vez presentado y defendido públicamente el trabajo en referencia hemos acordado: **APROBADO**.

En Guatamare, a los onces 11 días del mes de julio de 2017.

Lcdo. José Ordaz

C.I.: 10.165.822

Jurado Principal

MSc. Margarita Fajardo

C.I.:12.618.114

Jurado Principal

Lcdo. Alfredo Arabia

C.I.: 9.421.163

Jurado Principal

DEDICATORIAS

A Dios, a la Virgen Del Valle y a San Onofre Bendito.

A mi familia, por el apoyo incondicional que me ha brindado, en especial a mi abuela Gisela María, a mi Mama Mirna Josefina, a mis hermanas Yngrid Del Valle y Carelys Coromoto, mis sobrinos Sofía Del Valle, Gabriel Alejandro y Santiago Sebastián, gracias por toda su comprensión, apoyo y el cariño.

Patíño Blanco, Roger Aldayl.

AGRADECIMIENTOS

Al culminar la presente tesis, debo agradecer a:

- ✓ Dios todopoderoso, a quién me ha dado la fuerza para prosperar en nuestros estudios.
- ✓ Mi Familia, por habernos enseñado que con esfuerzo y dedicación todo se puede lograr. Su apoyo incondicional y su palabra amiga son invaluables.
- ✓ Mis profesores; Luis Rodríguez, Yvon La Rosa, Alfredo Arabia, por su contribución significativa en el duro camino del éxito académico.
- ✓ La Universidad de Oriente Núcleo Nueva Esparta, por habernos recibido con afecto y respeto durante nuestra formación académica.
- ✓ Mis compañeros de estudios en las carreras de Licenciatura en Informática, quienes nos acompañaron durante este arduo camino y nos apoyaron en todo momento para continuar adelante sin desfallecer.
- ✓ Todos aquellos que de forma solidaria han contribuido con nuestra exitosa formación académica.

ÍNDICE GENERAL

	Págs.
ACTA DE APROBACIÓN	ii
DEDICATORIA	iii
AGRADECIMIENTOS	iv
ÍNDICE GENERAL	v
LISTA DE TABLAS	viii
LISTAS DE FIGURAS	xi
RESUMEN	xii
INTRODUCCIÓN	xiii
CAPITULO I: MARCO INTRODUCTORIO	
Planteamiento del problema	16
Objetivos de la investigación	22
Objetivo general	22
Objetivos específicos	22
Justificación	23
CAPÍTULO II. MARCO TEÓRICO	
Antecedentes	24
Bases Teóricas	28
Lenguajes de Programación	28
PHP (Hipertext Pre-Preprocessor)	29
Frameworks	30
Manejador de Bases de Datos FirebirdSQL	30
HTTP	31
Servidor Web	31
REST	31
Códigos de estado HTTP	33
JSON (Notación de Objetos de JavaScript)	34
Tecnología móvil	34
Smartphone	35

Características de los Smartphone	36
Android	36
Núcleo Linux	37
Runtime	37
Librerías nativas	38
Entorno de aplicación	38
Notificaciones Push	39
Google Firebase	39
Firebase Cloud Messaging o FCM	39
Metodologías Ágiles de Desarrollo	40
Bases Legales	41
CAPÍTULO III. MARCO METODOLÓGICO	
Nivel de Investigación	43
Diseño y Tipo de Investigación	43
Técnicas de Recolección de Datos	44
Metodología de Desarrollo	46
Historias de usuarios	47
Roles	47
Programador	48
Cliente	48
Encargado de pruebas	48
Encargado de seguimiento	48
Entrenador	48
Consultor	49
Gestor	49
Procesos y práctica	49
Exploración	49
Planificación de la entrega	49
Iteraciones	50
Producción	50

Mantenimiento	50
Muerte del Proyecto	51
Prácticas de XP	51
CAPÍTULO IV. RESULTADOS	
Exploración	54
Planificación de la Entrega	63
Iteraciones	74
Producción	85
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES	
Conclusiones	104
Recomendaciones	106
REFERENCIAS	107
METADATOS	112

LISTA DE TABLAS

	Págs.
1. Historia de Usuario y Tareas	53
2. Iniciar sesión	55
3. Perfil	56
4. Imagen carnet	56
5. Dispositivo	57
6 Medidas	57
7. Notas	57
8. Apelaciones	58
9. Escuelas	58
10. Especialidades	58
11. Pensum	59
12. Electivas	59
13. Materia de un Pensum	59
14. Materias de un Semestre	60
15. Profesores	60
16. Reingresos	60
17. Semestres	61
18. Notificación	61
19. Constancias disponibles	62
20. Generar constancias	62
21. Aplicación móvil en Android	62
22. Web API REST	63
23. Desarrollo de ruta inicio de sesión	64
24. Desarrollo de ruta perfil de usuario	64
25. Desarrollo de ruta imagen de carnet	65
26. Desarrollo de ruta para almacenar dispositivo	65
27. Desarrollo de ruta para obtener medida	65
28. Desarrollo de ruta para obtener notas	66
29. Desarrollo de ruta para obtener apelaciones	66
30. Desarrollo de ruta para obtener escuelas	67

31. Desarrollo de ruta para obtener especialidades	67
32. Desarrollo de ruta para obtener pensum	67
33. Desarrollo de ruta para obtener electiva	68
34. Desarrollo de ruta materias de un pensum	68
35. Desarrollo de ruta materias de un semestre	69
36. Desarrollo ruta para obtener profesores	69
37. Desarrollo ruta para obtener reingresos	69
38. Desarrollo ruta para obtener semestres	70
39. Desarrollo ruta para enviar notificación	70
40. Desarrollo ruta para obtener semestres	71
41. Desarrollo ruta para generar constancias	71
42. Desarrollo de aplicación móvil en Android con las rutas del API REST	71
43. Desarrollo de página web para la documentación del API REST	72
44. Desarrollo de Diagrama de Gantt Unidimensional para el API REST	72
45. Desarrollo de Diagrama de Gantt Bidimensional para el API REST	73
46. Plan de entrega	75
47. Caso de prueba Ruta – Iniciar sesión	76
48. Caso de prueba Ruta – Perfil de usuario	76
49. Caso de prueba Ruta – Imagen de carnet	77
50. Caso de prueba Ruta – Almacenar dispositivo	77
51. Caso de prueba Ruta – Obtener medida	77
52. Caso de prueba Ruta – Notas	78
53. Caso de prueba Ruta – Apelación	78
54. Caso de prueba Ruta – Obtener Escuela	79
55. Caso de prueba Ruta – Especialidades	79
56. Caso de prueba Ruta – Pensum	79
57. Caso de prueba Ruta – Electiva	80

58. Caso de prueba Ruta – Materias de un pensum	80
59. Caso de prueba: Ruta – Materias de un semestre	81
60. Caso de prueba Ruta – Profesores	81
61. Caso de prueba Ruta – Reingreso	82
62. Caso de prueba Ruta – Semestre	82
63. Caso de prueba Ruta – Enviar Notificación	82
64. Caso de prueba Ruta – Constancias disponible	83
65. Caso de prueba Ruta – Generar constancias	83
66. Caso de prueba App Móvil	84
67. Caso de prueba Documentación del API REST	84
68. Definición de User-Agent del API REST	86
69. API REST Iniciar sesión	89
70. API REST Perfil	89
71. API REST Medidas	90
72. API REST Notas	90
73. API REST Apelación	91
74. API REST Escuelas	91
75. API REST Especialidad	92
76. API REST Pensum	92
77. API REST Materia	93
78. API REST Notificación	93
79. API REST Chechar Documento	94
80. API REST Generar Documento	94

LISTA DE FIGURAS

Págs.

1. Android Arquitectura	37
2. Iniciar sesión de aplicación móvil	96
3. Menú con opciones de aplicación móvil	96
4. Notificación usando la ruta API REST Notification	97
5. Perfil de aplicación móvil	97
6. Documentos de aplicación móvil	98
7. Especialidad de aplicación móvil	98
8. Pensum de aplicación móvil	99
9. Estructura de parámetros en rutas del API REST	99
10. Iniciar sesión en las rutas del API REST	100
11. Perfil en las rutas del API REST	100
12. Escuelas en las rutas del API REST	101
13. Apelación en las rutas del API REST	101
14. Especialidad en las rutas del API REST	102
15. Pensum en las rutas del API REST	102
16. Reingreso en las rutas del API REST	103



**UNIVERSIDAD DE ORIENTE
NÚCLEO NUEVA ESPARTA
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
LICENCIATURA EN INFORMÁTICA**

Desarrollo de una web API (Interfaz de programación de aplicaciones) con enfoque REST (Transferencia de Estado Representacional) para la gestión de Información Académica Estudiantil, como complemento al Sistema Académico En Línea (SAEL) de la Universidad de Oriente – Núcleo de Nueva Esparta.
Trabajo de Grado (Modalidad Investigación)

Autor(es): Br. Patiño Blanco, Roger Aldayl
Asesor Académico: Lcdo. Arabia, Alfredo
Fecha: Julio, 2017

RESUMEN

Actualmente, la Universidad de Oriente, no cuenta con un servicio web API REST o con algún otro mecanismo de desarrollo para obtener la información académica y estudiantil por medio de un cliente web o móvil. Es por ello, que la presente investigación, se basa en la creación de un servicio web API con arquitectura REST para obtener la data de la universidad, como complemento al sistema académico SAEL. Se utilizó la metodología ágil Extreme Programming (XP) de Kent Beck, también se utilizó el lenguaje programación PHP, Bases de Datos Firebird y para comprobar la operatividad de servicio Web, se implementó una aplicación móvil como prototipo y así presentar una nueva alternativa para realizar las diversas consultas académicas y la administración de información referente a la universidad como publicaciones, eventos, inscripciones o noticias mediante notificaciones push.

Descriptor(es): UDONE, SAEL, API, REST, ANDROID, XP, DOC.

INTRODUCCIÓN

El volumen de soluciones informáticas provenientes tanto desde el mundo del software libre como del sector comercial, bajo el formato de servicios web, se ha incrementado drásticamente durante los últimos años, ya que, los servicios web crecen con éxito gracias a los estándares creados para facilitar aquella integración. Tal como su nombre lo indica, los servicios web son aquellos que exponen una organización, empresa, o cualquier tipo de proveedor, para que sean utilizados por usuarios a través de la web o móviles, aprovechando todos los recursos que esta plataforma de intercomunicaciones entrega.

Dentro de los beneficios que aportan los servicios Web, aparte de la mencionada integración entre organizaciones independiente, está el hecho de que son fáciles de utilizar, tanto desde el punto de vista del proveedor de servicios como del consumidor. Ahora bien, esta ventaja está sujeta a los recursos que un servidor o computador tiene, pues la capacidad de procesamiento, disponibilidad de memoria y la velocidad de intercambio de información a través de la web, afectan la eficiencia de estos servicios.

La idea de exponer o consumir servicios web en dispositivo móviles pequeños, nace del importante uso que estos están teniendo en los últimos tiempos y, al incremento permanente de las prestaciones de dicho dispositivo. Además, se han creado estándares para facilitar la integración de servicios como las arquitecturas REST que consisten, fundamentalmente, en la identificación de los recursos que componen un sistema, concentrándose en proponer una ontología rica en sustantivos antes que en los verbos que operan sobre ellos.

Por lo tanto, las tecnologías web, siempre ha sido parte importante dentro de la sociedad y la educación, donde las universidades forman parte integral de este proceso, en el ámbito de la información y la

comunicación, debido a que utilizan sistemas y recursos para el desarrollo y difusión digitalizada de la información.

En la actualidad, se incorporan estas nuevas tecnologías al proceso educativo en sus distintos niveles con la finalidad de variar y flexibilizar las oportunidades de aprender sin restricciones de lugar, tiempo y atendiendo a las diferencias individuales y de grupo.

La Universidad de Oriente, específicamente el Núcleo de Nueva Esparta, se perfila en concordancia con estos nuevos avances en el área de la ciencia y la tecnología, por medio de la incorporación de nuevos proyectos enmarcados al desarrollo de aplicaciones usando plataforma Web, que permitirán dar a conocer y satisfacer las necesidades de información a toda la comunidad universitaria. UDONE desde el 2011 cuenta con un sistema web llamado Sistema Académico En Línea (SAEL), la cual permite que los estudiantes puedan inscribir sus materias siguiendo la hora de su cita de inscripción remotamente desde su casa, oficina, cibercafé u otro lugar, y así evitar que todos los estudiantes activos tengan que inscribirse dentro de las instalaciones de la universidad, lo que ocasiona grandes colas, errores a la hora de inscribir materias, etc. SAEL permite, además de la inscripción de asignaturas, realizar una solicitud de inscripción, aplicación de medidas académicas, solicitud de citas para carnet estudiantil, etc.

SAEL fue desarrollado desde sus inicios pensando en su uso desde navegadores web corriendo en sistemas de escritorio y no está optimizado para su uso a través de dispositivos móviles. SAEL obliga al estudiante a recurrir a sitios que posean un computador con acceso a internet para hacer uso de sus servicios.

El presente trabajo tiene por objetivo indagar acerca de la incidencia que tienen las aplicaciones basándose en el estándar REST, construyendo un servicio web API en conjunto con una aplicación móvil

como prototipo que permita obtener información académica de los estudiantes de la universidad, y así presentar una nueva alternativa para realizar las diversas consultas académicas y la administración de información referente a la universidad como publicaciones, eventos, inscripciones o noticias mediante notificaciones push utilizando la metodología ágil programación extrema (XP), el lenguaje de programación PHP, y el sistema manejador de Bases de Datos Firebird.

La presente investigación está dividida en cinco (5) capítulos, que se describen a continuación:

- Capítulo I. Marco Introdutorio; que se conforma del planteamiento del problema, objetivos de la investigación y justificación.
- Capítulo II. Marco Teórico; que enmarca los antecedentes y toda la teoría que rige las bases de esta investigación, incluyendo además las bases legales.
- Capítulo III. Marco Metodológico; donde se especifica el tipo y diseño de investigación, las técnicas de recolección de datos y la metodología de desarrollo aplicada.
- Capítulo VI. Resultados; en el cual se documenta la aplicación de la metodología XP en el desarrollo de la investigación.
- Capítulo V. Conclusiones y Recomendaciones; donde se describen las principales conclusiones a las que se llegaron con la investigación y se realizan humildes sugerencias para investigaciones futuras.

CAPÍTULO I

MARCO INTRODUCTORIO

PLANTEAMIENTO DEL PROBLEMA

El avance de las tecnologías de la información y comunicación ha alterado la cotidianidad de la sociedad y cambiado los modos de proceder e inclusive de vivir, tanto que según Pérez (2002) la sociedad actual es considerada la Sociedad de la información. Estas nuevas tecnologías han penetrado en todo tipo de ambientes y organizaciones, y su uso se ha hecho más intenso dadas las ventajas que aportan las redes y específicamente los servicios de Internet, en el acceso y manejo de gran cantidad de información y en la rápida transferencia de datos entre computadoras independientemente del lugar donde se encuentren.

Cada vez es más difícil encontrar instituciones, empresas, corporaciones, organizaciones independientes, compañía, un negocio familiar o las famosas “startup” o compañías de arranque que no pretendan aplicar soluciones informáticas para el desarrollo y difusión digitalizada de la información o para mejorar la gestión de su trabajo, al punto que la información y la tecnología necesaria para su manejo, se han convertido en valores estratégicos para las empresas y para sus gestores. Es por eso, que para maximizar la utilidad que posee la información, las organizaciones la deben manejar de forma correcta y eficiente, y de allí la importancia e impacto que tiene la implantación de sistemas de información en las organizaciones. (Quintero, 2002).

En la actualidad, muchas empresas brindan servicios a proveedores, personas o clientes para intercambiar y utilizar información, con el objetivo de ofrecer servicios mediante una interoperabilidad basada en la Web. Esta

interoperabilidad permite que las aplicaciones se comuniquen de forma independiente de la plataforma o de los lenguajes de programación usados en cada una de las aplicaciones o tecnologías conectadas entre sí. La arquitectura de un servicio web lo convierte en una interfaz de software que describe un conjunto de operaciones a las cuales se puede acceder por la red a través de protocolos estandarizados con el objetivo de describir una operación para ejecutar o datos para intercambiar con otro servicio web. Un grupo de servicios web que interactúa de esa forma definen aplicaciones web que, gracias al intercambio y uso de información o datos, simplifican al usuario hacer uso de distintos tipos de servicios, pudiendo estar estos creados y basados en plataformas totalmente distintas y geográficamente dispersas, con una misma apariencia o interface y para un mismo fin.

En definitiva, la tecnología avanza y es un hecho que conduce a que todos debemos estar actualizados y a la vanguardia, lo que indica mantenernos informados de lo que sucede a nuestro alrededor y más cuando nos referiremos e emplear sistemas información o aplicaciones móviles. En la actualidad, se incorporan estas nuevas tecnologías al proceso educativo en sus distintos niveles con la finalidad de variar y flexibilizar las oportunidades de aprender sin restricciones de lugar, tiempo y atendiendo a las diferencias individuales y de grupo. La Universidad de Oriente - Núcleo de Nueva Esparta (UDONE) se perfila en concordancia con estos nuevos avances en el área de la ciencia y la tecnología, por medio de la incorporación de nuevos proyectos enmarcados al desarrollo de aplicaciones usando plataforma Web, que permitirán satisfacer las necesidades de tecnologías de la información a toda la comunidad universitaria.

Desde el año 2011 la UDONE pone a disposición de la comunidad estudiantil una plataforma web llamada Sistema Académico En Línea (SAEL), la cual permite que los estudiantes puedan inscribir sus materias

siguiendo la hora de su cita de inscripción remotamente desde su casa, oficina, cibercafé u otro lugar, y así evitar que todos los estudiantes activos tengan que inscribirse dentro de las instalaciones de la universidad, lo que ocasiona grandes colas, errores a la hora de inscribir materias, etc. Según entrevista no estructurada al profesor Gustavo López, desarrollador y administrador principal de SAEL (enero 18, 2016).

El sistema SAEL permite, además de la inscripción de asignaturas, realizar una solicitud de inscripción por Listas de Espera, descargar documentos estudiantiles como constancias de inscripción, constancias de estudios, récord académico, solicitud de inscripción de Trabajo de Grado, Registro de Servicio Comunitario, solicitud de Retiro de Asignaturas, solicitud de Beneficios Estudiantiles, solicitud de Equivalencias Internas, visualizar los pensum de estudios, ver las materias aprobadas, ver los choques de materias antes de las inscripciones de las asignaturas, consultar la aplicación de medidas académicas, solicitud de citas para carnet estudiantil, entre otros.

El sistema SAEL se encuentra dividido en módulos, los cuales en su mayoría van dirigidos a la comunidad estudiantil; sin embargo, hay componentes específicos para el uso exclusivo de: personal del Área de Sistemas del Departamento de Admisión y Control de Estudios del Núcleo Nueva Esparta (DACENE), quienes fungen como administradores de SAEL; Jefes de Departamentos y Coordinadores de Programas; Representantes de Servicio Comunitario de las Carreras, Escuelas y Jefatura de Núcleo; personal de la Delegación de Desarrollo Estudiantil y Coordinador Académico del Núcleo.

El sistema SAEL fue desarrollado desde sus inicios pensando en su uso desde navegadores web corriendo en sistemas de escritorio y no está optimizado para su uso a través de dispositivos móviles, lo que hace que sea

molesto, tedioso e incómodo descargar planillas, constancias, subir la foto para el carnet, retirar materias desde un teléfono o Tablet, entre otros. Otra debilidad que presenta el sistema SAEL es que no cuenta con un diseño web adaptable o responsivo al tamaño de los distintos dispositivos existentes.

Bustamante (2011) indica que el diseño web responsivo es una serie de técnicas que permiten a una página web adaptarse al medio a través del cual un usuario está accediendo a la misma. Por otra parte, Mancera (2013) dice que el diseño web responsivo, se refiere a la adaptabilidad de una página en Internet hacia los diferentes equipos desde los cuales se puede acceder. Ejemplo: computador, televisión, laptop, tablet, smartphone, etc.

En concreto, la situación actual que presenta el sistema SAEL es que no cuenta con ningún tipo de servicio Web, API REST o algún mecanismo de desarrollo web basado en el estándar HTTP que le permita ser usado con comodidad por cualquier dispositivo o cliente móvil que entienda estos estándares.

ABC Tecnología (2015) señala que:

Una API (Interfaz de Programación de Aplicaciones) son un conjunto de comandos, funciones y protocolos informáticos que permiten a los desarrolladores crear programas específicos para ciertos sistemas operativos. Las API simplifican en gran medida el trabajo de un creador de programas, ya que no tiene que escribir códigos desde cero. Estas permiten al informático usar funciones predefinidas para interactuar con el sistema operativo o con otro programa.

Rafael Navarro (2006) explica que:

RESTFUL o REST (Transferencia de Estado Representacional) es un estilo de arquitectura de software para sistemas hipermedias distribuidos como la Web. Se refiere a una colección de principios para el diseño de arquitecturas en red. Estos principios resumen como los recursos son definidos y diseccionados. El término frecuentemente es utilizado en el sentido de describir a cualquier interfaz que transmite datos específicos de un domino sobre HTTP sin una capa adicional.

Por su parte, las API REST permiten identificar recursos que componen un sistema, concentrándose en proponer una ontología rica en sustantivos antes que en los verbos que operan sobre ellos. Una API REST, asociada directamente con la web, propone utilizar los métodos esenciales del protocolo HTTP: GET permite consultar y leer recursos, POST permite crear recursos, PUT permite editar recursos, DELETE permite eliminar recursos, PATCH permite editar partes concretas de un recurso.

Se debe agregar que, REST también permite usar formatos ligeros para el intercambio de datos como JSON (JavaScript Object Notation) que describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos; el cliente no le interesa en absoluto saber cómo está hecha la API REST, y el servidor no conoce que hará el cliente con los datos que le ha proporcionado. Otras de las funciones principales a la hora de implementar una API REST es permitir la posibilidad de crear tantos clientes como se necesiten para consumir la misma API REST, ya sea bajo plataforma web, web móvil responsiva, web de escritorio, escritorio con tecnología nativa, móvil nativo o móvil híbrido, lo cual reduce considerablemente el tiempo requerido para el desarrollo del lado del cliente (frontend).

Además, una API REST mejora la fiabilidad, escalabilidad, flexibilidad, se pueden hacer cambios en el servidor, en el lenguaje de desarrollo, en la base de datos, etc. Se puede desarrollar en cualquier tipo de tecnología o lenguaje dependiendo de las necesidades de lo que se va a desarrollar. Es indiferente que en el futuro se cambien totalmente las tecnologías con las que está implementada una API REST, siempre y cuando ésta siga teniendo las mismas operaciones, y cumpla las mismas funciones que se supone que debe hacer.

Al no estar optimizado para su uso desde móviles, el sistema SAEL obliga al estudiante a recurrir a sitios que posean un computador con acceso a internet para hacer uso de sus servicios, lo que demanda un costo en tiempo y dinero, y origina en los estudiantes una insatisfacción en el nivel de servicios de la información. El desarrollo y puesta en marcha a través de la presente propuesta de una API REST para SAEL, sentará las bases para el desarrollo acelerado de nuevas aplicaciones que puedan obtener toda la información de los estudiantes en formato JSON, sin importar si son aplicaciones móviles nativas, sitios web responsivos o una actualización tecnológica de la plataforma actual, que permitirán superar esta limitante de SAEL, y así brindar un mejor servicio a toda la Comunidad Estudiantil de la UDONE.

En vista de las deficiencias que existen en el Sistema Académico En Línea (SAEL) para su uso desde móviles, se propone la creación de una API REST usando el formato de intercambio de información JSON, esta propuesta contará con una serie de funciones basadas en el sistema actual, las cuales corresponden a los distintos módulos que son usados o visualizados por los estudiantes en el sistema SAEL, por otro lado esta investigación servirá como base para que futuros estudiantes interesados en el tema puedan seguir con el desarrollo e implementar nuevas funcionalidades o clientes.

La API REST estará alojada en el mismo servidor VPS (Servidor Virtual Privado) donde están ubicados los servicios web de la UDONE que se brindan actualmente a los estudiantes, como lo son: el Sistema Académico En Línea (SAEL), el Registro Único de Selección Interna (RUSI) y el Censo Estudiantil de Bachilleres de Nuevo Ingreso.

OBJETIVOS DE LA INVESTIGACIÓN

OBJETIVO GENERAL:

Desarrollo de una web API (Interfaz de programación de aplicaciones) con enfoque REST (Transferencia de Estado Representacional) para la gestión de Información Académica Estudiantil, como complemento al Sistema Académico En Línea (SAEL) de la Universidad de Oriente – Núcleo de Nueva Esparta.

OBJETIVO ESPECÍFICOS:

- Analizar las bases teóricas necesaria para la implementación de una web API con enfoque REST.
- Desarrollar una web API con enfoque REST utilizando la metodología ágil programación extrema (XP), el lenguaje de programación PHP, y el sistema manejador de Bases de Datos Firebird.
- Implementar un prototipo de aplicación móvil nativa bajo plataforma Android, para comprobar la correcta operatividad de algunas funciones de la API REST.
- Establecer una documentación para describir las funcionalidades y respuestas de las URI (Identificador de Recursos Uniforme) de las peticiones HTTP que admita la API REST.

JUSTIFICACIÓN

El presente trabajo de investigación es desarrollado con la finalidad de poder ofrecer a los estudiantes de la Universidad de Oriente - Núcleo Nueva Esparta, un servicio web API con enfoque REST en conjunto con una aplicación móvil como prototipo que permita obtener información académica de los estudiantes, y así presentar una nueva alternativa para realizar las diversas consultas académicas y la administración de información referente a la universidad como publicaciones, eventos, inscripciones o noticias mediante notificaciones push.

Una web API REST sería el primer modelo creado en UDONE, permitiendo reducir las barreras que existen para acceder a los recursos, información y poder establecer una conexión con los servidores privados de la universidad por vía del protocolo web HTTP y retornar la información que se desea. Poder desarrollar aplicaciones móviles que puedan ser instaladas en los teléfonos inteligentes, especialmente plataforma Android para poder realizar las mismas funciones del sistema SAEL de manera óptima desde un móvil, así lograr solventar algunos contratiempos que se presenta en sistemas actual, como obligar al estudiante a usar un computador, un navegador web, problemas externo de luz e internet a la hora se usar SAEL e insatisfacción en servicios en el nivel de la información.

La realización de esta investigación quedará como antecedentes en UDONE para que futuras generaciones, ya que; sentará bases para el desarrollo de los servicios web y poder acelerar el estudio de nuevos desarrollos de aplicaciones móviles que son claves importantes para el presente y futuro de las tecnologías en la actualidad.

CAPÍTULO II

MARCO TEÓRICO

ANTECEDENTES

Según Arias (2006). “Los antecedentes reflejan los avances y el Estado actual del conocimiento en un área determinada y sirven de modelo o ejemplo para futuras investigaciones”. Esta parte hace referencia a los estudios previos: trabajos y tesis de grado, artículos e informes científicos relacionados con el problema planteado, es decir, todas aquellas indagaciones realizadas anteriormente que puedan sustentar el estudio. Los antecedentes sirven de guía al investigador y le permiten hacer comparaciones y tener ideas sobre cómo se trató el problema en esa oportunidad.

Para dar respuesta al problema planteado en la siguiente investigación, se tomó como referencia los siguientes antecedentes:

Arce (2012). Planteó desarrollar en la Universidad de Mendoza (UM), de Provincia de Mendoza, Argentina, un trabajo de Tesis llamado: “Desarrollo e implementación de una API RESTful aplicada a un Sistema Académico Universitario” se trata de un desarrollo pionero en la Argentina:

El objetivo planteado en este trabajo de investigación consistió: en una aplicación web móvil multiplataforma que les posibilita a los estudiantes realizar actividades académicas desde sus Smartphone o teléfonos inteligentes. Dicho proyecto consta de dos etapas:

- ✓ La primera consiste en implementar un servicio Web RESTful orientado a recursos, que permite comunicar a la aplicación Web móvil con los datos de la Universidad.
- ✓ La segunda etapa, se basa en la implementación de una aplicación Web móvil multiplataforma, con la cual un alumno puede interactuar y realizar las operaciones permitidas por el servicio Web o la API. El servicio puede ser utilizado desde cualquier parte del mundo, sin necesidad de descargas y en cualquier teléfono. Los alumnos de la Universidad de Mendoza pueden utilizar este servicio tan sólo ingresando desde el navegador de sus teléfonos.

De igual manera, Vera (2012). "Diseño e implementación de una Web API para el Sistema Interactivo de Desarrollo para el Web (SIDWeb)". Informe de Proyecto de Graduación para optar por el título de Ingeniero en Ciencias Computacionales de la Escuela Superior Politécnica del Litoral (ESPOL), Guayaquil, Ecuador.

SIDWeb es un software de aprendizaje electrónico, de código abierto desarrollado por el Centro de Tecnologías de Información de la ESPOL. Este software es usado ampliamente por la comunidad de profesores y estudiantes para el manejo de contenidos, recursos adicionales de clases dictadas que facilitan la publicación de contenidos y la participación activa de los estudiantes de la universidad.

Por lo tanto, el objetivo principal fue implementar una API RESTful de comunicaciones que permita la interacción de aplicaciones externas móvil con el software SIDWeb a través de la API. Fue implementada con el framework SymfonyPHP y facilitó en gran medida el desarrollo de la API, el uso de un formato de transmisión con poca sobrecarga como JSON es

importante en interfaces como SIDWeb API, dado que el generar contenidos livianos lo hace más ventajoso, considerando factores críticos en aplicaciones móviles como el ancho de banda y la velocidad de procesamiento limitado.

Por otro lado, Pumarino (2010). Diseñó en la Universidad Católica de Chile, Santiago de Chile, un Trabajo de Grado denominado “Desarrollo de Plugins Distribuidos Como Servicios Web Rest”, desarrollando una aplicación basada en plugins distribuidos, servidos como servicios web bajo el estilo REST utilizando una Arquitectura Orientada a Recursos (ROA) y el framework de programación Ruby on Rails, para ser utilizada en la Fundación Educacional Loyola. Este proyecto ayuda a gestionar las donaciones que ingresan a la fundación a través de sistemas bancarios, el servicio web soluciona la automatización de ciertas tareas rutinarias que el administrador de la fundación debía realizar para concretar los mandatos de donaciones ordenados a través de los sistemas Pago Automático de Cuentas (PAC) del Banco de Chile y Pago Automático con Tarjeta (PAT) de Transbank. Estos plugins distribuidos como servicios web REST también pueden ser utilizados en programas de escritorio aprovechando todas las ventajas de una API REST.

Mientras que; Cambera (2009). Diseñó la aplicación “Composición de Servicios Web para Aplicaciones Móviles Geolocalizadas”, informe de pasantías realizado en la Universidad Simón Bolívar, como requisito parcial para optar al título de Ingeniero Electrónico. El objetivo de este trabajo fue desarrollar el diseño e implementación de una plataforma sencilla de composición de Servicios Web de interfaces tipo REST. Además de incorporar una interfaz Web final para su prueba en un dispositivo móvil iPhone. Dicho proyecto se dividió en cinco partes primordiales:

- ✓ Primero, desarrollo de las librerías y estructuras de datos para el manejo de peticiones a servicios Web, y respuestas de los mismos en formatos XML y JSON.
- ✓ Segundo, instalación de un servidor Tomcat 6.0, el cual junto con la implementación de Servlets permitía la comunicación con el cliente móvil.
- ✓ Tercero, implementación de una capa de presentación de servicios para dispositivos móviles basado en tecnologías Web.
- ✓ Cuarto, diseño de una base de datos MySQL que permitirá el caché de la información recientemente consultada.
- ✓ Quinto, puesta en marcha de un planificador de tareas para realizar actualizaciones periódicas de la base datos.

De acuerdo con; Chacón y Teodoro (2010). “Diseño De Una Aplicación Móvil Para La Consulta Académica De La Facultad de Ingeniería Industrial y de Sistemas (FIIS)”, en la Universidad Tecnológica del Perú; dicha aplicación permite la consulta de horarios, la verificación de asistencias, detalle de las calificaciones tanto de prácticas calificadas como de exámenes parciales y finales, así como el control de pagos y su respectiva fecha de vencimiento, etc. Hay que mencionar también, que el desarrollo de la aplicación fue construida con el sistema operativo Android para teléfonos inteligentes y tabletas, además, se utilizó Lenguaje Unificado de Modelado (UML) permitiendo generar diseños y lógicas de los módulos de la aplicación.

BASES TEÓRICAS

Según Bavaresco (2006) las bases teóricas tiene que ver con las teorías que brindan al investigador el apoyo inicial dentro del conocimiento del objeto de estudio, es decir, cada problema posee algún referente teórico, lo que indica, que el investigador no puede hacer abstracción por el desconocimiento, salvo que sus estudios se soporten en investigaciones puras o bien exploratorias.

Lenguajes de Programación

Un lenguaje de programación es un sistema notacional para describir computaciones de una forma legible tanto para la máquina como para el ser humano. (Louden, 2002).

Según se aproximen al lenguaje nativo del ordenador, llamado código máquina o a la forma humana de expresar órdenes, se distinguen distintos niveles de lenguajes de programación:

- a) Lenguajes de Bajo nivel: son los más rápidos y ponen al servicio del programador todo el equipo físico. Se utilizan para el desarrollo de programas básicos (otros lenguajes, control de procesos, etc.). Como ejemplo puede citarse el ensamblador.
- b) Lenguajes de alto nivel: en este grupo se encuentran los más conocidos. La elaboración de programas es más sencilla y cada lenguaje suele estar enfocado a un determinado campo. Entre ellos los más habituales son: COBOL, FORTRAN, BASIC, PASCAL, C, PROLOG, MUMPS.
- c) Lenguajes de cuarta generación: son las herramientas de desarrollo de programas. En este caso las órdenes existentes están mucho más próximas a la forma de pensar humana y por ello resulta más sencillo,

flexible y productivo trabajar con estos lenguajes. Estas herramientas suelen constar de generadores de pantallas, generadores de informes, generadores de aplicaciones y utilidades. Como ejemplos más habituales pueden citarse: Oracle, Transtool, Clipper, Informix, DBase IV, etc.

- d) Otros lenguajes: el gran auge de los entornos Windows ha conducido al desarrollo de lenguajes visuales que utilizan la programación orientada a objetos. Entre estos lenguajes está Visual Basic de Microsoft, Delphi de Borland, C++ y SQL.

PHP (Hiptertext Pre-Preprocessor)

PHP es un lenguaje de desarrollo web escrito por y para los desarrolladores web. PHP significa: Hipertext Pre-Preprocessor. El producto fue originalmente llamado Personal Home Page Tools, Actualmente se encuentra en su séptima reescritura, llamado PHP7 o simplemente PHP.

Además, Maestro de la web (2001) explica que PHP, es un scripts que corre en el servidor, que puede ser embebido en HTML o usado únicamente como binario (aunque el uso anterior es mucho más común). PHP es un lenguaje ideal para aprender a desarrollar aplicaciones web, que suele ser utilizado conjuntamente con servidores Apache y NGINX este último más popular, bases de datos SQL (Structured Query Language) como MySQL, PostgreSQL, SQLite y FirebirdSQL y bases de datos NoSQL como MongoDB, CouchDB, Apache Cassandra y en sistemas Windows, Mac y Linux, permitiendo combinaciones barata (todos los componentes son de código libre), potente y versátil.

Frameworks

Gutiérrez (2014), comenta que un framework como una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Por lo tanto, los Frameworks son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software, que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. Además, fuera de las aplicaciones en la informática, un framework puede ser considerado como el conjunto de procesos y tecnologías usados para resolver un problema complejo y, es el esqueleto sobre el cual varios objetos son integrados para una solución dada.

Manejador de Bases de Datos FirebirdSQL

Bringas (2000), explica que Firebird es un sistema de administración de bases de datos relacional (Lenguaje consultas SQL) de código abierto, basado en la versión 6 de Interbase. Así mismo, Firebird cuenta con un rendimiento excelente y se escala de manera impresionante, desde un modelo integrado y monousuario, hasta desarrollos empresariales con múltiples bases de datos de más de 500 GB, con cientos de clientes simultáneos.

Firebird soporta un número grande de plataformas de software y hardware: Windows, Linux, MacOS, HP-UX, AIX, Solaris y más. Éste funciona en x386, x64 y PowerPC, Sparc y otras plataformas de hardware, además, cuenta con un mecanismo de fácil migración entre tales

plataformas. Una de las características claves de Firebird es su arquitectura multigeneracional, que permite el desarrollo y soporte de aplicaciones híbridas, soporta procedimientos almacenados, disparadores, eventos y funciones definidas por el usuario.

HTTP

Es un Protocolo de transferencia de hipertexto y, es el mecanismo más extendido utilizado por los servidores y los navegadores para comunicarse. Aparte, este protocolo regula las solicitudes de los navegadores y transferencia de fichero HTML, soportando la recuperación y presentación de texto, grafico, animaciones y la reposición de sonidos. A su vez, HTTP maneja múltiples protocolo de aplicaciones permitiendo al usuario acceder a muchos servicios, tales como FTP, Goher, WAIS, debido a que el protocolo HTTP es la base de la mayor parte de transacciones web, los servidores web HTTP se llaman también servidores HTTP. (Auxiliares administrativo de la Comunidad Autónoma de País Vasco, 2005).

Servidor Web

Un servidor web es un programa que gestiona cualquier aplicación al lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente, generando una respuesta en cualquier lenguaje o aplicación. Por lo tanto, el código recibido por el cliente suele ser compilado y ejecutado por un Navegador Web, utilizado para la transmisión de todos estos datos algún protocolo. Generalmente, se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del Modelo OSI. (EcuRed, 2009)002E

REST

BBVAOpen4U (2016) menciona, que REST es una técnica arquitectónica sin estado. Esto quiere decir, que los recursos donde se realizan las peticiones únicamente disponen de una imagen del sistema formado por el recurso hacia el que realizar la petición a través del conector correspondiente. Dicho de otro modo, una petición realizada desde el cliente al servidor, es independiente de las siguientes peticiones. Esto implica que una aplicación, a través de un identificador de recurso y la operación que implemente sobre ella, accede a dicho recurso, abstrayéndose del resto de componentes o recursos que forman en el sistema, la información que es devuelta debe ser proporcionada en el formato correspondiente, ya sea este XML, JSON, JPEG o HTML, etc. Los contenido de servicios web REST son cacheables, esto es, una vez realizada una primera petición, el resto de peticiones se apoyara en la cache, en caso de que lo necesite.

Por otro lado, en la actualidad, existen empresas como: Twitter, YouTube, Facebook, Github, Google, entre otros que generan negocio gracias a las APIs REST. Sin ellas, todo el crecimiento en horizontal sería prácticamente imposible. Esto es así porque REST es el estándar más lógico, eficiente y habitual en la creación de APIs para servicios de Internet.

Por lo tanto, BBVAOpen4U (2016) describes las características de REST como las siguientes:

- ✓ Protocolo cliente / servidor sin estado: cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla. Aunque esto es así, algunas aplicaciones HTTP incorporan memoria caché. Se configura lo que se conoce como protocolo cliente-caché-servidor sin estado: existe la posibilidad de

definir algunas respuestas a peticiones HTTP concretas como cacheables, con el objetivo de que el cliente pueda ejecutar en un futuro la misma respuesta para peticiones idénticas. De todas formas, que exista la posibilidad no significa que sea lo más recomendable.

- ✓ Las operaciones más importantes relacionadas con los datos en cualquier sistema REST y la especificación HTTP son cuatro: POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar).
- ✓ Los objetos en REST siempre se manipulan a partir de la URI. Es la URI y ningún otro elemento el identificador único de cada recurso de ese sistema REST. La URI nos facilita acceder a la información para su modificación o borrado, por ejemplo, para compartir su ubicación exacta con terceros.
- ✓ Interfaz uniforme: para la transferencia de datos en un sistema REST, este aplica acciones concretas (POST, GET, PUT y DELETE) sobre los recursos, siempre y cuando estén identificados con una URI. Esto facilita la existencia de una interfaz uniforme que sistematiza el proceso con la información.
- ✓ Sistema de capas: arquitectura jerárquica entre los componentes. Cada una de estas capas lleva a cabo una funcionalidad dentro del sistema REST.

Códigos de estado HTTP

Lázaro (2016), declara que, los códigos de estado HTTP describen de forma abreviada la respuesta HTTP. Estos códigos están especificados por el RFC 2616. El primer dígito del código de estado especifica uno de los 5 tipos de respuesta, el mínimo para que un cliente pueda trabajar con HTTP es que reconozca estas 5 clases. La Internet Assigned Numbers Authority (IANA) mantiene el registro oficial de códigos de estado HTTP.

Los códigos son:

- ✓ 1XX Respuestas informativas.
- ✓ 2XX Peticiones correctas.
- ✓ 3XX Redirecciones.
- ✓ 4XX Errores del cliente.
- ✓ 5XX Errores de servidor.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript)

La Ecma International (1999), describe JSON como un formato ligero de intercambio de datos. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

Tecnología móvil

Según Miguel (2008), “los teléfonos móviles han dejado de ser un mero instrumento de comunicación interpersonal”. Si bien el grueso de los usuarios aún no ha abrazado todas las posibilidades abiertas por extensión de la nueva generación de redes y el desarrollo de terminales “inteligentes”, estos dispositivos y sus aplicaciones ya forman parte “de nuestra vida cotidiana, de los espacios de interacción social y de los ritos cotidianos en los que construimos como sujetos y sociedades” en un grado suficiente como para explorar en profundidad su potencial y el modo en que los usuarios los asumen.

Con lo mencionado anteriormente, un Smartphone con los distintos Sistemas Operativo tiene como base la tecnología móvil que es exactamente lo que el nombre implica, una tecnología que es portátil. Los dispositivos móviles puede ser capaces de utilizar una variedad de tecnologías de la comunicación, tales como:

- ✓ Fidelidad inalámbrica (WI-FI), tecnología inalámbrica de red ubicada en una determinada área.
- ✓ Bluetooth, conecta dispositivos móviles de forma inalámbrica.
- ✓ “Tercera Generación” (3G o UMTS), sistema global para comunicaciones móviles (GSM) y el servicio general de radio paquetes (GPRS), servicios de datos, los datos de los servicios de redes para teléfonos móviles.
- ✓ H de HSPDA: High Speed Downlink Packet Access, también conocida como 3.5G, 3G+ o Turbo3G, es la optimización de la tecnología espectral UMTS/WCDMA, incluida en las especificaciones de 3GPP Release 5 y consiste en un nuevo canal compartido en el enlace descendente (Downlink) que mejora significadamente la capacidad máxima de transparencia de información pudiéndose alcanzar tasas de hasta 14 Mbps de descargar.
- ✓ “Cuarta Generación” (4G LTE o Long Term Evolution / evolución a largo plazo), son las siglas para referirse a la cuarta generación de tecnologías de telefonía Móvil. Su principal diferencia frente a las tecnologías anteriores, es la rapidez en el acceso a los servicios a través de las redes. 4G LTE permite velocidades 5 veces mayores a las de HSPA+ (4G, 3.5G), 10 veces más que 3G y 250 veces más que 2G.
- ✓ Redes privadas virtuales, un acceso seguro a una red privada,

Smartphone

El Smartphone conocido como teléfono inteligente, son dispositivos electrónicos que se asemejan a un teléfono celular, es como una computadora personal pero móvil, que permite la comunicación como un celular normal. (Quiminet, 2011).

Características de los Smartphone

- ✓ Se usa para enviar correos electrónicos.
- ✓ Contiene GPS para no perderse jamás y ubicar cualquier dirección.
- ✓ Permite la instalación de otros programas.
- ✓ Permite cualquier interfaz para el ingreso de datos.
- ✓ Tiene agenda digital y administra los contactos.
- ✓ Se pueden leer documentos con diferentes formatos.
- ✓ Permite el ingreso a internet.
- ✓ Cuentan con un sistema operativo.
- ✓ Permiso de instalación de programas de terceros.
- ✓ Realización, edición y almacenamiento de videos y fotos.
- ✓ Reproductor mp3.

Android

Universidad Carlos III de Madrid, (2012), define Android como un sistema operativo y una plataforma software, basado en Linux para teléfonos móviles. Además, también usan este sistema operativo (aunque no es muy habitual), tablets, netbooks, reproductores de música e incluso PC. Android permite programar en un entorno de trabajo (framework) de Java, aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución). Además, lo que le diferencia de otros sistemas operativos, es que cualquier persona que sepa programar puede crear nuevas aplicaciones, widgets, o incluso, modificar el

propio sistema operativo, dado que Android es de código libre, por lo que sabiendo programar en lenguaje Java, va a ser muy fácil comenzar a programar en esta plataforma.

Hay que mencionar, que Android es una plataforma software con el objetivo de abstraer el hardware y facilitar el desarrollo de aplicaciones para dispositivos. A continuación, observamos en la Figura 1, como de forma esquematizada aparecen los componentes principales del sistema:

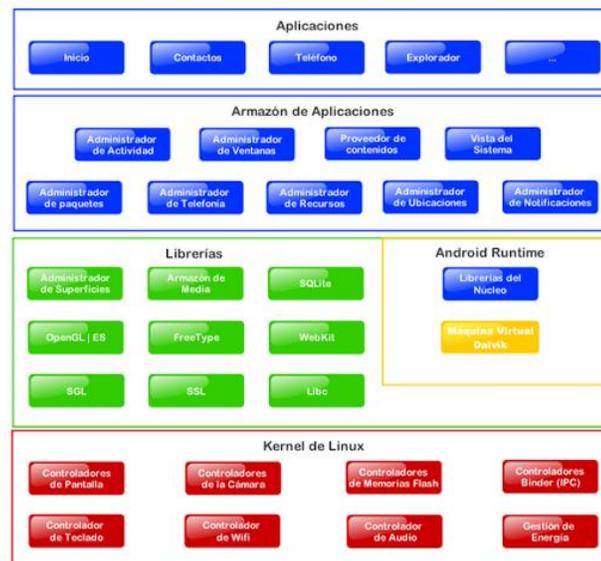


Figura 1. Universidad Carlos III de Madrid. (2012). Android Arquitectura.

Núcleo Linux

El núcleo de Android está formado por el sistema operativo Linux versión 2.6. Esta capa proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de drivers para dispositivos. Dicha capa del modelo, actúa como capa de abstracción del hardware, por lo tanto, es la única que es dependiente del hardware. (Gironés, 2013).

Runtime

Está basado en el concepto de máquina virtual utilizado en Java pero dadas las limitaciones de los Android, Google tomó la decisión de crear una nueva, la máquina virtual Dalvik, que respondiera mejor a estas limitaciones y, esta optimizada para ahorrar la máxima memoria posible trabajando con registros y delegando operaciones al Kernel. El Runtime también contiene la mayoría de las librerías disponibles en el lenguaje Java. (Gironés, 2013).

Librerías nativas

Incluye un conjunto de librerías en C/C++ usadas en varios componentes de Android que están compiladas en código nativo del procesador. (Gironés, 2013).

Entorno de aplicación

Proporciona una plataforma de desarrollo libre para aplicaciones con gran riqueza e innovaciones lo que permite a los desarrolladores implementar en sus aplicaciones funcionalidades como ubicación, notificaciones y el uso de todos los sensores. (Gironés, 2013).

El entorno de la aplicación, contiene servicios que permiten a las aplicaciones ser ejecutadas y mostradas a los usuarios además de darle funcionalidades.

Los servicios más importantes que incluye son:

1. Views: extenso conjunto de vistas, (parte visual de los componentes).
2. Resource Manager: proporciona acceso a recursos que no son en código. Cualquier recurso (imagen, audio,...) que no se encuentre dentro de la aplicación necesita llamar a este manejador.

3. Activity Manager: maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
4. Notification Manager: permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
5. Content Providers: mecanismo sencillo para acceder a datos de otras aplicaciones (como los contactos).

Notificaciones Push

Según Revelo (2016), las push notifications o notificaciones push son paquetes de datos enviados desde un servidor hacia sus clientes sin necesidad de una petición previa. Una Notificación push es un aviso, mensaje o notificación, que lanza un sistema y que se recibe en un dispositivo móvil a través de la aplicación o programa correspondiente. Al mismo tiempo, es una manera de comunicación a través de Internet, entre una aplicación que hace las veces de servidor, y una aplicación que hace las veces de cliente. La petición de un intercambio o transferencia se inicia en el servidor, no siendo necesario que el cliente haga una petición para comprobar si existe algo nuevo. El servidor siempre lo notifica.

Google Firebase

Google (2016). Define Google Firebase como una plataforma nueva de backend para el desarrollo móvil en la nube. Está disponible para diferentes plataformas (Android, iOS, Web). Firebase provee varias API para guardar y sincronizar datos en la nube en tiempo real. Entre sus características principales incluye autenticación de usuarios y almacenamiento (hosting) estático, monetiza de aplicaciones, análisis de datos, informes de fallos de las aplicaciones, envió de notificaciones push etc.

Firestore Cloud Messaging o FCM

Es un servicio de Google cuya función popular es el envío de notificaciones push desde una aplicación servidor hacia una aplicación cliente. Fue creado para aplicarse a escala y actualmente hace que lleguen 170 mil millones de mensajes por día a 2 mil millones de dispositivos. (Google, 2016).

Esto es posible gracias a la intervención del servicio de mensajería del servidor de Firestore, está disponible para uso gratuito e ilimitado, admite mensajes en iOS, Android y plataformas web, y está altamente optimizada para brindar confiabilidad y eficacia en el consumo de batería.

Metodologías Ágiles de Desarrollo

Según Vera (2013). La Metodología Ágil es un marco metodológico de trabajo que plantea permitir mejorar la eficiencia en la producción y la calidad de los productos finales, tener la capacidad de respuesta al cambio en los productos y sus definiciones, y brindar la mayor satisfacción posible al cliente, a través de la entrega temprana y la retroalimentación continua durante la construcción del producto.

Entre las metodologías ágiles más destacadas podemos nombrar:
Metodologías

- XP – Extreme Programming.
- Scrum.
- Kanban.

BASES LEGALES

Según Villafranca (2002). “Las bases legales no son más que se leyes que sustentan de forma legal el desarrollo del proyecto” explica que las bases legales “son leyes, reglamentos y normas necesarias en algunas investigaciones cuyo tema así lo amerite”.

A continuación se mencionan (2) artículos del Decreto N° 3390 Publicado en la Gaceta oficial N° 38.095 de fecha 28/12/2004, de Fecha: 23 de diciembre de 2004, los cuales se consideraron para el desarrollo de la presente investigación:

Artículo1. La Administración Pública Nacional empleará prioritariamente Software Libre desarrollado con Estándares Abiertos, en sus sistemas, proyectos y servicios informáticos. A tales fines, todos los órganos y entes de la Administración Pública Nacional iniciarán los procesos de migración gradual y progresiva de éstos hacia el Software Libre desarrollado con Estándares Abiertos.

Artículo 2. A los efectos del presente Decreto se entenderá por: Software Libre: Programa de computación cuya licencia garantiza al usuario acceso al código fuente del programa y lo autoriza a ejecutarlo con cualquier propósito, modificarlo y redistribuir tanto el programa original como sus modificaciones en las mismas condiciones de licenciamiento acordadas al programa original, sin tener que pagar regalías a los desarrolladores previos.

Estándares Abiertos: Especificaciones técnicas, publicadas y controladas por alguna organización que se encarga de su desarrollo, las cuales han sido aceptadas por la industria, estando a disposición de cualquier usuario para ser implementadas en un software libre u otro, promoviendo la competitividad, interoperatividad o flexibilidad.

Software Propietario: Programa de computación cuya licencia establece restricciones de uso, redistribución o modificación por parte de los usuarios, o requiere de autorización expresa del Licenciador.

Distribución Software Libre desarrollado con Estándares Abiertos para el Estado Venezolano: Un paquete de programas y aplicaciones de Informática elaborado utilizando Software Libre con Estándares Abiertos para ser utilizados y distribuidos entre distintos usuarios.

De acuerdo, con los artículos antes expuestos, el proyecto a desarrollar, se ajusta a los requerimientos, ya que el Infogobierno se encuentra en el deber de desarrollar e implementar aplicaciones utilizando Software Libre.

Por lo tanto, la aplicación propuesta, será desarrollada utilizando el lenguaje de programación PHP 5, Framework FlightPHP para desarrollo de las rutas de las API REST, Sistema Operativo Android y el manejador de bases de datos FirebirdSQL, ya que son tecnologías de información libres y cumplen con los principios legales establecidos.

Capítulo III Marco Metodológico

Nivel de Investigación

El nivel de una Investigación según Arias (1999). “Se refiere al grado de profundidad con que se aborda un objeto o fenómeno”. (p.19). Por el tipo de investigación, el estudio se ubica dentro de un Nivel Comprensivo, el cual Hurtado (2000), lo define como aquel donde "se estudia el evento en su relación con otros eventos, dentro de un holos mayor, enfatizando por lo general las relaciones de causalidad, aunque no exclusivamente" (p.23). Este nivel viene dado por el objetivo de la Investigación establecido, el cual busca proponer un cambio, basado en los requerimientos reales de la Institución objeto de estudio.

Cuando se trata de resolver un problema, es muy conveniente tener un conocimiento detallado de los posibles tipos de investigación que se puedan seguir. Según Hurtado (2000), se establece que, para dar respuestas a los diferentes objetivos de una investigación, ésta debe estar enmarcada en los criterios que permitan definir la manera cómo se recolecta la información, lo cual sirve de base para la delimitación o demarcación de la investigación.

Diseño y Tipo de Investigación

El diseño de investigación según Tamayo y Tamayo (1997, p. 70) “es la estructura a seguir en una investigación, ejerciendo el control de la misma

a fin de encontrar resultados confiables y su relación con las interrogantes surgidas del problema”.

Mientras que, Hurtado (2010, p. 132), indica que “los tipos de investigación en este caso se definen por el objetivo, más que por el área de conocimiento en la cual se realiza el estudio”. Entonces, el tipo de investigación señala el grado de profundidad y el tipo de resultado, en concordancia con el objetivo general, y el diseño tiene que ver con los procedimientos específicos para recoger los datos (fuentes, tiempo y cantidad de eventos de estudio).

En este sentido, dentro de los tipos de investigación, está la investigación proyectiva, que de acuerdo a Hurtado (2010, p. 133), es la investigación que tiene como objetivo diseñar o crear propuestas dirigidas a resolver determinadas situaciones. Los proyectos de arquitectura e ingeniería, el diseño de maquinarias, la creación de programas de intervención social, el diseño de programas de estudio, los inventos, la elaboración de programas informáticos, entre otros, siempre que estén sustentados en un proceso de investigación, son ejemplos de investigación proyectiva.

Es por eso que la presente investigación encaja en una investigación proyectiva, ya que consiste en la elaboración de una propuesta, usando una tecnología de gran auge en la actualidad como lo es una web API REST, que permitirá que terceros puedan desarrollar de forma acelerada aplicaciones móviles acopladas al Sistema Académico En Línea (SAEL) de la Universidad de Oriente – Núcleo de Nueva Esparta.

Técnicas de Recolección de Datos

El desarrollo de todo trabajo investigativo requiere de la aplicación de técnicas e instrumentos diseñados para la recolección de datos o información que permitan llegar a la óptima solución del problema planteado. Hurtado (2000) indica que: "La selección de técnicas e instrumentos de recolección de datos implica determinar por cuáles medios o procedimientos el investigador obtendrá la información necesaria para alcanzar los objetivos de la investigación". (p.164).

Por lo tanto, para el desarrollo de este proyecto las técnicas empleadas en la recolección de los datos serán, en primer lugar, la observación directa la cual consiste en el contacto directo con los elementos o características del hecho o fenómeno que se pretende investigar. Según Sabino (1997), señala que: "La observación directa es aquella a través de la cual se puedan conocer los hechos y situaciones de la realidad social". (p. 134).

Otra de las técnicas utilizadas fue la entrevista no estructurada, la cual trata de la realización de preguntas abiertas sin ningún orden preestablecido, donde las interrogantes van surgiendo en base a las respuestas que va dando el entrevistado. En este sentido, Sabino (2002), comenta que la entrevista no estructurada es aquella en que existe un margen grande de libertad para formular las preguntas y respuestas. (p. 108).

En este caso, se fundamentará en los diálogos y conversaciones informales con algunos estudiantes de la universidad, y con las personas que se involucran en la investigación como el Prof. Gustavo López, desarrollador y administrador de SAEL, obteniendo información necesaria para completar la presente propuesta.

METODOLOGÍA DE DESARROLLO

Las metodologías ágiles de desarrollo impulsan una gestión de proyectos que promueve el trabajo en equipo, de forma auto organizada, organización y responsabilidad propia, junto con un conjunto de buenas prácticas de ingeniería de software, las cuales brindan una entrega rápida de software y de alta calidad, junto con un enfoque de negocios que alinea el desarrollo con las necesidades del cliente y los objetivos de la compañía (LNCS, 2009).

La metodología usada para el desarrollo del proyecto, Desarrollo de una web API (Interfaz de Programación de Aplicaciones) con enfoque REST (Transferencia de Estado Representacional) para la gestión de Información Académica Estudiantil, como complemento al Sistema Académico En Línea (SAEL) de la Universidad de Oriente – Núcleo de Nueva Esparta, fue la programación extrema o Extreme Programming (XP), el proyecto que se desarrolló tomo como base algunas funcionalidades sitio web principal SAEL, ejemplos de algunas funcionalidades son como iniciar sesión, obtener datos personales o perfil, notas, reingreso, medidas, etc. Estas funcionalidades pueden ser tomadas como historias de usuarios. Se contó con la participación directa del Lcdo. Gustavo López que se puede conocer como un programador más, debido a que ayudo en algunas funcionalidades del API REST por razones de seguridad de la universidad.

La metodología XP (Extreme Programming) fue creada por Kent Beck en el año 1996 y está basada en sus años de práctica como un desarrollador

de software orientado a objeto. Las raíces de XP yacen en la comunidad de Smalltalk, y en particular de la colaboración de Kent Beck junto a Ward Cunningham a finales de la década de los años ochenta.

Así mismo, Beck (2000) define XP (Extreme Programming) como:

Una metodología ágil para el desarrollo de software destinada a ser utilizada por equipos de desarrollo pequeños y medianos (de 2 a 10 miembros) que se enfrenten a proyectos con requerimientos imprecisos o cambiantes. Las relaciones desarrollador - desarrollador y desarrolladores - cliente son fundamentales en esta metodología. La adopción del cliente como un miembro más del equipo de desarrollo es la clave del éxito.

Esta metodología XP plantea un conjunto de valores y prácticas, que permite a los desarrolladores dedicarse a lo que hacen mejor, programar. Esta metodología elimina la necesidad de dedicar tiempo a labores tediosas y burocráticas, establecidas para los procesos no ágiles, tales como: exhaustivos documentos de proyecto, diagrama Gantt, enormes volúmenes de listas de requerimientos, juntas de revisiones interminables, entre otros.

Por otro lado; Beck (2000) dice que la programación XP se caracteriza por organizarse en tres diferentes puntos: historias de usuario, roles, proceso y prácticas:

En primer lugar, Historias de Usuario: son fichas donde el cliente describe las características del sistema. Las historias de usuario son dinámicas y altamente flexibles, esto con el fin de poder cambiar o reemplazar las ideas escritas con la mayor facilidad posible, con la finalidad de ejecutarla en semanas. La información dentro de la historia de los usuarios es variable, no existe un estándar de cómo deben ser escritas, por lo tanto, al comienzo de cada iteración, se registrarán los cambios en las historias de usuario y según eso se planificará la siguiente iteración.

En segunda instancia, se debe agregar que XP, como toda metodología para el desarrollo de software, propone varios roles. Cada uno acarrea consigo ciertas responsabilidades; algunas son menos complejas que otras.

- ✓ **Programador:** es una pieza clave en XP. Su responsabilidad no se limita a implementar cierta funcionalidad del sistema; él también debe comunicarse, ya sea con otros miembros del equipo de desarrollo o con el cliente, elaborar pruebas unitarias y llevar a cabo las integraciones del sistema.
- ✓ **Cliente:** el cliente debe escribir las historias de usuario y las pruebas funcionales del sistema, asignarles prioridad a las historias de usuarios y tomar decisiones acerca de cuál se debe implementar en cada iteración; en fin, convertirse en un miembro más del equipo de desarrollo.
- ✓ **Encargado de pruebas:** verifica que el sistema esté funcionando correctamente. Entre los deberes del encargado de pruebas se encuentran ejecutar regularmente todos los casos de prueba, informar al equipo los resultados obtenidos y ayudar al cliente a escribir las pruebas funcionales del sistema. En caso de utilizarse alguna herramienta de soporte para pruebas, ésta se encontrará bajo su responsabilidad.
- ✓ **Encargado de seguimiento:** es la conciencia del equipo de desarrollo. Debe verificar el cumplimiento del plan de entrega y del plan de iteración, e informar si las estimaciones realizadas fueron

correctas, se subestimó o sobrestimó; con el objetivo de que el equipo sea más preciso en futuras estimaciones.

- ✓ **Entrenador:** es quien advierte si ocurre una desviación en el proceso, mantiene la calma cuando todos se encuentran “aterrados”, en fin, guía al equipo de desarrollo para que se siga el proceso XP correctamente.
- ✓ **Consultor:** es responsable de guiar al equipo de desarrollo para resolver los problemas que se les presente en un tema específico. El equipo de desarrollo en ocasiones necesita conocimientos de un tema, el cual no domina ninguno de sus miembros.
- ✓ **Gestor:** es la máxima autoridad del equipo de desarrollo. Él debe poseer ciertas cualidades como coraje, confianza y en ocasiones insistencia sobre los miembros del equipo para que se realice el trabajo, pero no debe agobiarlos.

Finalmente, los procesos y prácticas.

- ✓ **Exploración:**
 1. Los desarrolladores y el cliente elaboran la metáfora.
 2. El cliente redacta las historias de usuario.
 3. Los desarrolladores dividen las historias de usuario en tareas de programación y calculan los puntos estimados de cada historia de usuario.
 4. Los desarrolladores estudian las tecnologías a utilizar.
 5. Los desarrolladores construyen uno o varios prototipos del sistema.
 6. Los desarrolladores realizan una propuesta inicial de la arquitectura del sistema.

7. El cliente redacta los casos de prueba ayudado por el encargado de pruebas.

✓ **Planificación:**

8. El cliente asigna prioridad a las historias de usuario.

9. Los desarrolladores estiman el esfuerzo total.

10. Los desarrolladores estiman la velocidad del equipo.

11. El cliente y los desarrolladores elaboran el plan de entrega.

12. Actualizar documentación.

✓ **Iteraciones:**

13. El cliente reasigna prioridad a las historias de usuario.

14. El cliente y los desarrolladores elaboran el plan de iteración.

15. Los desarrolladores definen el diseño preliminar.

16. El cliente y los desarrolladores mejoran los casos de prueba.

17. Los programadores comienzan la implementación.

18. Los desarrolladores mejoran el diseño.

19. El cliente y los desarrolladores completan y actualizan el plan de iteración.

20. Actualizar documentación.

✓ **Producción:**

21. Completar y actualizar el plan de entrega.

22. El cliente y los desarrolladores se reúnen diariamente para informar en que continúa trabajando cada cual.

23. El cliente y los desarrolladores realizan continuas pruebas al sistema.

✓ **Mantenimiento:**

24. El cliente y los desarrolladores comienzan una nueva entrega, con lo que se regresa a la fase de exploración. Por lo tanto, luego de ser puesta en producción una primera versión del sistema, se desarrollan las nuevas iteraciones, con la finalidad de realizar tareas de soporte para el cliente, esta fase puede requerir nuevo personal dentro del equipo y cambios en su estructura.

✓ **Muerte del proyecto:**

25. Así mismo, esta fase llega cuando el cliente no posee más historias para ser incluidas en el sistema. Además, esto se logra cuando son satisfechas las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto.

Por lo tanto, las prácticas de XP:

✓ **El juego de Planeamiento:** Rápidamente determinar el alcance del próximo release mediante la combinación de prioridades del negocio y estimaciones técnicas. A medida que la realidad va cambiando el plan, actualizar el mismo.

✓ **Pequeños Releases:** Poner un sistema simple en producción rápidamente, luego liberar nuevas versiones en ciclos muy cortos.

✓ **Metáfora:** Guiar todo el desarrollo con una historia simple y compartida de cómo funciona todo el sistema.

- ✓ **Diseño Simple:** El sistema deberá ser diseñado tan simple como sea posible en cada momento. Complejidad extra es removida apenas es descubierta.
- ✓ **Testing:** Los programadores continuamente escriben pruebas unitarias, las cuales deben correr sin problemas para que el desarrollo continúe. Los clientes escriben pruebas demostrando que las funcionalidades están terminadas.
- ✓ **Refactoring:** Los programadores reestructuran el sistema sin cambiar su comportamiento para remover duplicación, mejorar la comunicación, simplificar, o añadir flexibilidad.
- ✓ **Programación de a Pares:** Todo el código de producción es escrito por dos programadores en una máquina (no aplicable a este proyecto).
- ✓ **Propiedad Colectiva del Código:** Cualquiera puede cambiar código en cualquier parte del sistema en cualquier momento.
- ✓ **Integración Continua:** Integrar y hacer builds del sistema varias veces por día, cada vez que una tarea se completa.
- ✓ **Semana de 40 horas:** Trabajar no más de 40 horas semanales como regla. Nunca trabajar horas extras durante dos semanas consecutivas.
- ✓ **Cliente en el lugar de Desarrollo:** Incluir un cliente real en el equipo, disponible de forma full-time para responder preguntas.

- ✓ **Estándares de Codificación:** Los programadores escriben todo el código de acuerdo con reglas que enfatizan la comunicación a través del mismo.

Tabla 1. Serrano (2012). Historia de Usuario y Tareas

Historia de Usuario	
Número: 1	Nombre: Enviar artículo
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 2
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: (Alto / Medio / Bajo)	Puntos Reales:
Descripción: Se introducen los datos del artículo (título, fichero adjunto, resumen, tópicos) y de los autores (nombre, e-mail, afiliación).	
Observaciones:	

CAPITULO IV RESULTADOS

En el proyecto Desarrollo de una web API (Interfaz de Programación de Aplicaciones) con enfoque REST (Transferencia de Estado Representacional) para la gestión de Información Académica Estudiantil, como complemento al Sistema Académico En Línea (SAEL) de la Universidad de Oriente – Núcleo de Nueva Esparta, se implementó la metodología ágil Programación Extrema (XP, eXtreme Programming por sus siglas en inglés). El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

De acuerdo con los objetivos de esta investigación y con lo que expresa cada una de las fases de la metodología XP, se llegó a la conclusión de desarrollar sólo las siguientes fases: Exploración, Planificación de la entrega, Iteración y Producción, esto debido a que el proyecto debe estar implantado y en funcionamiento para ejecutar las etapas de Mantenimiento y Muerte del proyecto, para así poder complementar o agregar las diversas

historias que surjan y satisfacer las necesidades del cliente en otros aspectos como rendimiento y confiabilidad.

A continuación, se presentan las fases aplicadas de la metodología de desarrollo XP:

Fase I. Exploración

Mediante entrevista no estructurada realizada al Lcdo. Gustavo López (Desarrollador y Administrador del sistema SAEL), se lograron obtener datos importantes como determinadas tablas de la base de datos de SAEL, durante esta fase de exploración se determinaron los requerimientos para el proyecto, entre ellos se encuentran:

1. Iniciar sesión y generar un token de autenticación para tener acceso al API REST.
2. Mostrar el perfil de usuario y datos como cedula, nombres, apellidos, promedio general, etc.
3. Obtener imagen de carnet estudiantil.
4. Enviar información del dispositivo móvil, como modelo, versión del sistema operativo, clave para el envío de notificaciones, etc.
5. Mostrar las medidas académicas que le han sido aplicadas a un estudiante.
6. Mostrar en pantalla el récord académico de un estudiante.
7. Mostrar las apelaciones a medidas académicas.
8. Mostrar: escuelas, especialidades, pensum de estudios, materias electivas, profesores.
9. Mostrar los reingresos de un estudiante.
10. Enviar notificaciones push a través de Google Firebase.
11. Obtener constancias disponibles de estudiante y el tipo de constancia.

12. Generar en PDF: constancia de inscripción, de estudios, récord académico.

13. Aplicación móvil en Android para probar el funcionamiento del API REST.

14. Web para mostrar la información sobre el uso de las rutas del API REST.

A continuación, se mostrarán a grandes rasgos las historias de usuario (ver Tabla 2 al 22) que fueron de interés para la primera entrega del proyecto.

Tabla 2. Iniciar sesión

Historia de Usuario	
Número: 1	Usuario: Desarrollador
Nombre historia: Iniciar sesión	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 0.7
Descripción: Permitir que un cliente que entienda http pueda iniciar sesión y generar un token de autenticación de usuario.	
Observaciones: El usuario debe estar debidamente registrado en la plataforma SAEL desde la web.	

Tabla 3. Perfil

Historia de Usuario	
Número: 2	Usuario: Desarrollador
Nombre historia: Perfil	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 0.2
Descripción: Obtener información personal del estudiante como nombres, apellidos, promedio general, carrera actual, etc.	
Observaciones:	

Tabla 4. Imagen carnet

Historia de Usuario	
Número: 3	Usuario: Desarrollador
Nombre historia: Imagen carnet	
Prioridad en negocio: Baja	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 0.1
Descripción: Obtener imagen de carnet universitario	
Observaciones:	

Tabla 5. Dispositivo

Historia de Usuario	
Número: 4	Usuario: Desarrollador
Nombre historia: Dispositivo	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 1
Descripción: Enviar información del dispositivo móvil, como modelo, versión del sistema operativo, clave para el envío de notificaciones, etc.	
Observaciones: Se requiere iniciar sesión primero.	

Tabla 6. Medidas

Historia de Usuario	
Número: 5	Usuario: Desarrollador
Nombre historia: Medidas	

Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 0.4
Descripción: Obtener las medidas académica aplicadas al estudiante.	
Observaciones:	

Tabla 7. Notas

Historia de Usuario	
Número: 6	Usuario: Desarrollador
Nombre historia: Notas	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 1
Descripción: Obtener las notas que tiene el estudiante.	
Observaciones:	

Tabla 8. Apelaciones

Historia de Usuario	
Número: 7	Usuario: Desarrollador
Nombre historia: Apelaciones	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 0.2
Descripción: Obtener las apelaciones a medidas académicas que tiene el estudiante.	
Observaciones:	

Tabla 9. Escuelas

Historia de Usuario

Número: 8	Usuario: Desarrollador
Nombre historia: Escuelas	
Prioridad en negocio: Media	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 0.1
Descripción: Obtener las escuelas registradas en la universidad.	
Observaciones:	

Tabla 10. Especialidades

Historia de Usuario	
Número: 9	Usuario: Desarrollador
Nombre historia: Especialidades	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 0.2
Descripción: Obtener las especialidades registradas en la universidad.	
Observaciones:	

Tabla 11. Pensum

Historia de Usuario	
Número: 10	Usuario: Desarrollador
Nombre historia: Pensum	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 1.2
Descripción: Obtener el pensum de estudios de una especialidad.	
Observaciones: Se requiere el código de especialidad	

Tabla 12. Electivas

Historia de Usuario	
Número: 11	Usuario: Desarrollador
Nombre historia: Electivas	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 1
Descripción: Obtener las materias electivas de una especialidad.	
Observaciones: Se requiere el código de especialidad.	

Tabla 13. Materia de un Pensum

Historia de Usuario	
Número: 12	Usuario: Desarrollador
Nombre historia: Materia de un Pensum	
Prioridad en negocio: Media	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 0.4
Descripción: Obtener materia de un pensum.	
Observaciones:	

Tabla 14. Materias de un Semestre

Historia de Usuario	
Número: 13	Usuario: Desarrollador
Nombre historia: Materias de un Semestre	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 1
Descripción: Obtener las materias de un pensum en específico por semestre.	
Observaciones: Se requiere el código de una especialidad y el número de semestre.	

Tabla 15. Profesores

Historia de Usuario	
Número: 14	Usuario: Desarrollador
Nombre historia: Profesores	
Prioridad en negocio: Baja	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 0.1
Descripción: Obtener los profesores que están registrados en la universidad.	
Observaciones:	

Tabla 16. Reingresos

Historia de Usuario	
Número: 15	Usuario: Desarrollador
Nombre historia: Reingresos	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 0.3
Descripción: Obtener los reingresos que tiene aprobados el estudiante.	
Observaciones:	

Tabla 17. Semestres

Historia de Usuario	
Número: 16	Usuario: Desarrollador
Nombre historia: Semestres	
Prioridad en negocio: Baja	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 0.2
Descripción: Obtener los semestres, semestre con año y periodo que han pasado en la universidad.	

Observaciones:

Tabla 18. Notificación

Historia de Usuario	
Número: 17	Usuario: Desarrollador
Nombre historia: Notificación	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 2.1
Descripción: Implementación para el envío de notificaciones push desde el API Rest a los dispositivos móviles	
Observaciones: Solo para administradores	

Tabla 19. Constancias disponibles

Historia de Usuario	
Número: 18	Usuario: Desarrollador
Nombre historia: Constancias disponibles.	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 0.5
Descripción: Obtener las constancias disponibles para un estudiante.	
Observaciones:	

Tabla 20. Generar constancias

Historia de Usuario	
Número: 19	Usuario: Desarrollador
Nombre historia: Generar constancias	

Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 0.8
Descripción: Generar constancias de estudio, inscripción y record académico.	
Observaciones: Se tiene que verificar que el estudiante pueda generar constancia de estudios, récord académico.	

Tabla 21. Aplicación móvil en Android

Historia de Usuario	
Número: 20	Usuario: Usuario
Nombre historia: Aplicación móvil en Android	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 12
Descripción: Desarrollo de un prototipo de aplicación móvil.	
Observaciones: Se requiere por completo las rutas del API REST	

Tabla 22. Web API REST

Historia de Usuario	
Número: 21	Usuario: Desarrollador
Nombre historia: Web API REST	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Baja	Puntos estimados: 0.3
Descripción: Web para mostrar la información sobre el uso de las rutas API REST.	
Observaciones: Se requiere por completo las rutas del API REST	

Fase II. Planificación de la Entrega

En esta fase se realizó la planificación de las distintas tareas que se llevaron a cabo. Para ello existen una serie de criterios que hay que seguir para que las partes implicadas en este proceso (desarrollador y cliente) tengan voz y se sientan parte de la decisión tomada, las cuales son:

1. Cada módulo de programación se realizaba de manera rápida, con patrones de diseño, eficiente y con pruebas; por lo tanto, el empleo de cada característica, dependiendo de su finalidad, era lograda en poco tiempo.
2. El horario de desarrollo del investigador (Roger Patiño) era de 06:00 PM a 11:00 PM, 09:00 AM a 08:00 PM o 01:00PM a 11:00 PM, de lunes a lunes.
3. Cada historia posee un intervalo entre uno (1), dos (2), tres (3) y hasta noventa (90) días aproximadamente para su realización, esto dio una noción inicial de cuánto tiempo se tardaría en escribir el API y ponerla a pruebas mediante un prototipo de aplicación móvil y la creación de la documentación web de las rutas del API REST, para realizarse con éxito. En breve se mostrarán todas las tareas realizadas (ver Tablas 23 al 43) y los Diagramas de Gantt Unidimensional (ver Tabla 44) y Bidimensional (ver Tabla 45), implementados para el desarrollo del proyecto.

Tabla 23. Desarrollo de ruta inicio de sesión

Tarea	
Número tarea: 1	Número historia: 1
Nombre de la tarea: Desarrollo de ruta inicio de sesión	
Tipo de tarea: Desarrollo	Puntos estimados: 0.7
Fecha de Inicio: 01/06/2016	Fecha de Finalización: 05/06/2016

Programador responsable: Roger A. Patiño B.
Descripción: Se obtiene los datos de usuario como cedula y contraseña, si los datos son correctos se genera un token de autenticación permanente o parcial de 10 minutos.
Observaciones: Se requiere cedula y contraseña en formato JSON

Tabla 24. Desarrollo de ruta perfil de usuario

Tarea	
Número tarea: 2	Número historia: 2
Nombre de la tarea: Desarrollo de ruta perfil de usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.7
Fecha de Inicio: 06/06/2016	Fecha de Finalización: 07/06/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Se imprime el perfil de usuario que está almacenado en el servidor en formato JSON	
Observaciones: Se requiere el token de autenticación.	

Tabla 25. Desarrollo de ruta imagen de carnet

Tarea	
Número tarea: 3	Número historia: 3
Nombre de la tarea: Desarrollo de ruta imagen de carnet	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de Inicio: 07/06/2016	Fecha de Finalización: 07/06/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Se obtiene la ubicación de la imagen de carnet imprimir en formato JSON	
Observaciones: Se requiere el token de autenticación.	

Tabla 26. Desarrollo de ruta para almacenar dispositivo

Tarea	
Número tarea: 4	Número historia: 4
Nombre de la tarea: Desarrollo de ruta para almacenar dispositivo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de Inicio: 08/06/2016	Fecha de Finalización: 14/06/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Almacenar la información del dispositivo que inició sesión, como dispositivo móvil, modelo, versión del sistema operativo, clave para el envío de notificaciones, etc. La información tiene que estar en formato JSON	
Observaciones: Se requiere el token de autenticación.	

Tabla 27. Desarrollo de ruta para obtener medida

Tarea	
Número tarea: 5	Número historia: 5
Nombre de la tarea: Desarrollo de ruta para obtener medida	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha de Inicio: 15/06/2016	Fecha de Finalización: 18/06/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Se obtiene las medidas académicas y enviarlas en formato JSON	
Observaciones: Se requiere el token de autenticación.	

Tabla 28. Desarrollo de ruta para obtener notas

Tarea	
Número tarea: 6	Número historia: 6
Nombre de la tarea: Desarrollo de ruta para obtener notas	
Tipo de tarea: Desarrollo	Puntos estimados: 1

Fecha de Inicio: 19/06/2016	Fecha de Finalización: 25/06/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Obtener medidas académicas y enviarlas en formato JSON	
Observaciones: Se requiere el token de autenticación.	

Tabla 29. Desarrollo de ruta para obtener apelaciones

Tarea	
Número tarea: 7	Número historia: 7
Nombre de la tarea: Desarrollo de ruta para obtener apelaciones	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de Inicio: 26/06/2016	Fecha de Finalización: 27/06/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Obtener apelaciones y enviarlas en formato JSON	
Observaciones: Se requiere el token de autenticación.	

Tabla 30. Desarrollo de ruta para obtener escuelas

Tarea	
Número tarea: 8	Número historia: 8
Nombre de la tarea: Desarrollo de ruta para obtener escuelas	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de Inicio: 01/07/2016	Fecha de Finalización: 01/07/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Obtener escuelas y enviarlas en formato JSON	
Observaciones: Se requiere el token de autenticación.	

Tabla 31. Desarrollo de ruta para obtener especialidades

Tarea	
Número tarea: 9	Número historia: 9
Nombre de la tarea: Desarrollo de ruta para obtener especialidades	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de Inicio: 02/07/2016	Fecha de Finalización: 03/07/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Obtener especialidades y enviarlas en formato JSON	
Observaciones: Se requiere el token de autenticación.	

Tabla 32. Desarrollo de ruta para obtener pensum

Tarea	
Número tarea: 10	Número historia: 10
Nombre de la tarea: Desarrollo de ruta para obtener pensum	
Tipo de tarea: Desarrollo	Puntos estimados: 1.2
Fecha de Inicio: 04/07/2016	Fecha de Finalización: 12/07/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Obtener pensum y enviarlas en formato JSON	
Observaciones: Se requiere el token de autenticación y código de especialidad	

Tabla 33. Desarrollo de ruta para obtener electiva

Tarea	
Número tarea: 11	Número historia: 11
Nombre de la tarea: Desarrollo de ruta para obtener electiva	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de Inicio: 13/07/2016	Fecha de Finalización: 19/07/2016

Programador responsable: Roger A. Patiño B.
Descripción: Obtener electiva y enviarlas en formato JSON
Observaciones: Se requiere el token de autenticación y código de especialidad

Tabla 34. Desarrollo de ruta materias de un pensum

Tarea	
Número tarea: 12	Número historia: 12
Nombre de la tarea: Desarrollo de ruta materia de un pensum	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha de Inicio: 20/07/2016	Fecha de Finalización: 23/07/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Obtener materia de un pensum y enviarlas en formato JSON	
Observaciones: Se requiere el token de autenticación, código de especialidad, identificador de pensum y código de materia	

Tabla 35. Desarrollo de ruta materias de un semestre

Tarea	
Número tarea: 13	Número historia: 13
Nombre de la tarea: Desarrollo de ruta materias de un semestre	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de Inicio: 24/07/2016	Fecha de Finalización: 30/07/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Obtener materias de un pensum y semestre en específico y enviarlas en formato JSON	
Observaciones: Se requiere el token de autenticación, código de especialidad, identificador de pensum, semestre en número.	

Tabla 36. Desarrollo ruta para obtener profesores

Tarea	
Número tarea: 14	Número historia: 14
Nombre de la tarea: Desarrollo ruta para obtener profesores	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de Inicio: 01/08/2016	Fecha de Finalización: 01/08/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Obtener profesores y enviarlas en formato JSON	
Observaciones: Se requiere el token de autenticación.	

Tabla 37. Desarrollo ruta para obtener reingresos

Tarea	
Número tarea: 15	Número historia: 15
Nombre de la tarea: Desarrollo ruta para obtener reingresos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha de Inicio: 02/08/2016	Fecha de Finalización: 04/08/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Obtener reingresos y enviarlas en formato JSON	
Observaciones: Se requiere el token de autenticación.	

Tabla 38. Desarrollo ruta para obtener semestres

Tarea	
Número tarea: 16	Número historia: 16
Nombre de la tarea: Desarrollo ruta para obtener semestres	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2

Fecha de Inicio: 05/08/2016	Fecha de Finalización: 06/08/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Obtener semestres y enviarlas en formato JSON	
Observaciones: Se requiere el token de autenticación.	

Tabla 39. Desarrollo ruta para enviar notificación

Tarea	
Número tarea: 17	Número historia: 17
Nombre de la tarea: Desarrollo ruta para enviar notificación	
Tipo de tarea: Desarrollo	Puntos estimados: 2.1
Fecha de Inicio: 07/08/2016	Fecha de Finalización: 21/08/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Enviar notificación compuesto por título, descripción, fecha, cédulas, especialidades y enviarlas al servidor en formato JSON	
Observaciones: Se requiere el token de autenticación.	

Tabla 40. Desarrollo ruta para obtener semestres

Tarea	
Número tarea: 18	Número historia: 18
Nombre de la tarea: Desarrollo ruta para constancias disponibles	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de Inicio: 22/08/2016	Fecha de Finalización: 26/08/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Verificar si el usuario puede generar constancias	
Observaciones: Se requiere el token de autenticación.	

Tabla 41. Desarrollo ruta para generar constancias

Tarea	
Número tarea: 19	Número historia: 19
Nombre de la tarea: Desarrollo ruta para generar constancias	
Tipo de tarea: Desarrollo	Puntos estimados: 0.8
Fecha de Inicio: 27/08/2016	Fecha de Finalización: 02/09/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Generar constancias de estudio, inscripción y record académico.	
Observaciones: Se requiere el token de autenticación, tipo de constancias y la cantidad de constancias.	

Tabla 42. Desarrollo de aplicación móvil en Android con las rutas del API REST

Tarea	
Número tarea: 20	Número historia: 20
Nombre de la tarea: Desarrollo aplicación móvil en Android	
Tipo de tarea: Desarrollo	Puntos estimados: 12
Fecha de Inicio: 03/09/2016	Fecha de Finalización: 03/11/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Desarrollo de aplicación móvil con las rutas del API REST	
Observaciones: Se requiere por completo las rutas del API REST	

Tabla 43. Desarrollo de página web para la documentación del API REST

Tarea	
Número tarea: 21	Número historia: 21
Nombre de la tarea: Desarrollo de documentación del API REST	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3

Fecha de Inicio: 04/11/2016	Fecha de Finalización: 06/11/2016
Programador responsable: Roger A. Patiño B.	
Descripción: Desarrollo página web para documentación del API REST	
Observaciones: Se requiere por completo las rutas del API REST	

Tabla 44. Desarrollo de Diagrama de Gantt Unidimensional para el API REST

	Actividad	Duración (Días)	Fecha inicial	Fecha final
1	Desarrollo de ruta inicio de sesión	5	01/06/2016	05/06/2016
2	Desarrollo de ruta perfil de usuario	2	06/06/2016	07/06/2016
3	Desarrollo de ruta imagen de carnet	1	07/06/2016	07/06/2016
4	Desarrollo de ruta para almacenar dispositivo	7	08/06/2016	14/06/2016
5	Desarrollo de ruta para obtener medida	4	15/06/2016	18/06/2016
6	Desarrollo de ruta para obtener notas	7	19/06/2016	25/06/2016
7	Desarrollo de ruta para obtener apelaciones	2	26/06/2016	27/06/2016
8	Desarrollo de ruta para obtener escuelas	1	01/07/2016	01/07/2016
9	Desarrollo de ruta para obtener especialidades	2	02/07/2016	03/07/2016
10	Desarrollo de ruta para obtener pensum	9	04/07/2016	12/07/2016
11	Desarrollo de ruta para obtener electiva	7	13/07/2016	19/07/2016
12	Desarrollo de ruta materia de un pensum	4	20/07/2016	23/07/2016
13	Desarrollo de ruta materias de un semestre	7	24/07/2016	30/07/2016
14	Desarrollo ruta para obtener profesores	1	01/08/2016	01/08/2016
15	Desarrollo ruta para obtener reingresos	3	02/08/2016	04/08/2016

16	Desarrollo ruta para obtener semestres	2	05/08/2016	06/08/2016
17	Desarrollo ruta para enviar notificación	15	07/08/2016	21/08/2016
18	Desarrollo ruta para constancias disponibles	5	22/08/2016	26/08/2016
19	Desarrollo ruta para generar constancias	6	27/08/2016	02/09/2016
20	Desarrollo aplicación móvil en Android	90	03/09/2016	03/11/2016
21	Desarrollo de documentación de API Rest	3	04/11/2016	06/11/2016

Tabla 45. Desarrollo de Diagrama de Gantt Bidimensional para el API REST

Mes	Junio				Julio				Agosto				Septiembre				Octubre				Noviembre			
Semana	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	█																							
2																								
3	█																							
4	█	█																						
5			█																					
6			█	█																				
7			█																					
8				█																				
9				█																				
10				█	█																			
11					█	█																		
12							█																	
13								█																

2	No existía	0.1	1 días	No existía	No existía
3	No existía	0.1	1 días	No existía	No existía
4	No existía	1	7 días	No existía	No existía
5	No existía	0.2	2 días	No existía	No existía
6	No existía	1	7 días	No existía	No existía
7	No existía	0.2	2 días	No existía	No existía
8	No existía	0.1	1 días	No existía	No existía
9	No existía	0.1	1 días	No existía	No existía
10	No existía	1	7 días	No existía	No existía
11	No existía	0.7	5 días	No existía	No existía
12	No existía	0.5	4 días	No existía	No existía
13	No existía	1	7 días	No existía	No existía
14	No existía	0.1	1 días	No existía	No existía
15	No existía	0.4	3 días	No existía	No existía
16	No existía	0.1	1 días	No existía	No existía
17	No existía	1.5	11 días	No existía	No existía
18	No existía	0.7	5 días	No existía	No existía
19	No existía	0.8	6 días	No existía	No existía
20	No existía	12	90 días	No existía	No existía
21	No existía	0.4	3 días	No existía	No existía

Para verificar el código desarrollado se realizaron casos de prueba de las distintas tareas que se fueron definidas (ver Tabla 47 al 67).

Tabla 47. Caso de prueba Ruta – Iniciar sesión

Caso de prueba

Número de Caso de Prueba: 1	Número de Historia: 1
Nombre de Caso de Prueba: Ruta – Iniciar sesión	
Descripción: Iniciar sesión en el API REST	
Condiciones de ejecución: Obtener datos de los estudiantes en la base de datos de UDONE, para ser validados.	
Entrada/Salida: Cedula y contraseña en formato JSON, el Token de autenticación en formato JSON	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 48. Caso de prueba Ruta – Perfil de usuario

Caso de prueba	
Número de Caso de Prueba: 2	Número de Historia: 2
Nombre de Caso de Prueba: Ruta – Perfil de usuario	
Descripción:	
Condiciones de ejecución: Se debe enviar el Token de autenticación para obtener el perfil.	
Entrada/Salida: Token de autenticación, perfil de usuario en formato JSON	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 49. Caso de prueba Ruta – Imagen de carnet

Caso de prueba	
Número de Caso de Prueba: 3	Número de Historia: 3
Nombre de Caso de Prueba: Ruta – Imagen de carnet	
Descripción:	
Condiciones de ejecución: Se debe enviar el Token de autenticación para obtener la imagen de perfil	

Entrada/Salida: Token de autenticación, url de la ubicación de la imagen de perfil en servidor en formato JSON
Resultados esperados: No se reportó ningún error de ejecución
Evaluación: Buena

Tabla 50. Caso de prueba Ruta – Almacenar dispositivo

Caso de prueba	
Número de Caso de Prueba: 4	Número de Historia: 4
Nombre de Caso de Prueba: Ruta – Almacenar dispositivo	
Descripción: Almacenar la información de dispositivo que inicio sesión	
Condiciones de ejecución: Se requieren datos del dispositivo móvil	
Entrada/Salida: Modelo del dispositivo, versión del sistema operativo, clave para el envío de notificaciones, placa base del dispositivo, numero de compilación del sistema móvil en formato JSON, código indicado si se almaceno correctamente	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 51. Caso de prueba Ruta – Obtener medida

Caso de prueba	
Número de Caso de Prueba: 5	Número de Historia: 5
Nombre de Caso de Prueba: Ruta – Obtener medida	
Descripción:	
Condiciones de ejecución: Se debe enviar el Token de autenticación para obtener medidas	
Entrada/Salida: Token de autenticación, medidas de usuario en formato JSON	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 52. Caso de prueba Ruta – Notas

Caso de prueba	
Número de Caso de Prueba: 6	Número de Historia: 6
Nombre de Caso de Prueba: Ruta - Notas	
Descripción:	
Condiciones de ejecución: Se debe enviar el Token de autenticación para obtener notas	
Entrada/Salida: Token de autenticación, notas de usuario en formato JSON	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 53. Caso de prueba Ruta – Apelación

Caso de prueba	
Número de Caso de Prueba: 7	Número de Historia: 7
Nombre de Caso de Prueba: Ruta – Apelación	
Descripción:	
Condiciones de ejecución: Se debe enviar el Token de autenticación para obtener apelaciones	
Entrada/Salida: Token de autenticación, apelaciones de usuario en formato JSON	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 54. Caso de prueba Ruta – Obtener escuela

Caso de prueba	
Número de Caso de Prueba: 8	Número de Historia: 8
Nombre de Caso de Prueba: Ruta – Obtener escuela	
Descripción:	

Condiciones de ejecución: Se debe enviar el Token de autenticación para obtener escuelas
Entrada/Salida: Token de autenticación, escuelas de UDONE en formato JSON
Resultados esperados: No se reportó ningún error de ejecución
Evaluación: Buena

Tabla 55. Caso de prueba Ruta – Especialidades

Caso de prueba	
Número de Caso de Prueba: 9	Número de Historia: 9
Nombre de Caso de Prueba: Ruta - Especialidades	
Descripción:	
Condiciones de ejecución: Se debe enviar el Token de autenticación para obtener escuelas	
Entrada/Salida: Token de autenticación, especialidades de UDONE en formato JSON	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 56. Caso de prueba Ruta – Pensum

Caso de prueba	
Número de Caso de Prueba: 10	Número de Historia: 10
Nombre de Caso de Prueba: Ruta – Pensum	
Descripción:	
Condiciones de ejecución: Se debe enviar el Token de autenticación e identificador de la especialidad	
Entrada/Salida: Token de autenticación, identificador de especialidad, pensum de especialidad en formato JSON	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 57. Caso de prueba Ruta – Electiva

Caso de prueba	
Número de Caso de Prueba: 11	Número de Historia: 11
Nombre de Caso de Prueba: Ruta - Electiva	
Descripción:	
Condiciones de ejecución: Se debe enviar el Token de autenticación e identificador de la especialidad	
Entrada/Salida: Token de autenticación, identificador de especialidad, electivas de especialidad en formato JSON	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 58. Caso de prueba Ruta – Materias de un pensum

Caso de prueba	
Número de Caso de Prueba: 12	Número de Historia: 12
Nombre de Caso de Prueba: Ruta - Materias de un pensum	
Descripción: Obtener materia de un pensum de una especialidad	
Condiciones de ejecución: Se requiere token de autenticación, identificador de especialidad, identificador de pensum y código de materia	
Entrada/Salida: Token de autenticación, identificador de especialidad, identificador de pensum, código de materia, materia en formato JSON	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 59. Caso de prueba: Ruta – Materias de un semestre

Caso de prueba	
Número de Caso de Prueba: 13	Número de Historia: 13

Nombre de Caso de Prueba: Ruta – Materias de un semestre
Descripción:
Condiciones de ejecución: Se requiere token de autenticación, identificador de especialidad, identificador de pensum y numero de semestre
Entrada/Salida: Token de autenticación, identificador de especialidad, identificador de pensum, numero del semestre, materias en formato JSON
Resultados esperados: No se reportó ningún error de ejecución
Evaluación: Buena

Tabla 60. Caso de prueba Ruta – Profesores

Caso de prueba	
Número de Caso de Prueba: 14	Número de Historia: 14
Nombre de Caso de Prueba: Ruta - Profesores	
Descripción:	
Condiciones de ejecución: Se debe enviar el Token de autenticación para obtener profesores	
Entrada/Salida: Token de autenticación, datos de profesores de UDONE en formato JSON	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 61. Caso de prueba Ruta – Reingreso

Caso de prueba	
Número de Caso de Prueba: 15	Número de Historia: 15
Nombre de Caso de Prueba: Ruta – Reingreso	
Descripción:	
Condiciones de ejecución: Se debe enviar el Token de autenticación para obtener reingresos	

Entrada/Salida: Token de autenticación, reingresos de usuario en formato JSON
Resultados esperados: No se reportó ningún error de ejecución
Evaluación: Buena

Tabla 62. Caso de prueba Ruta – Semestre

Caso de prueba	
Número de Caso de Prueba: 16	Número de Historia: 16
Nombre de Caso de Prueba: Ruta – Semestre	
Descripción:	
Condiciones de ejecución: Se debe enviar el Token de autenticación para obtener los semestres de UDONE	
Entrada/Salida: Token de autenticación, semestre de UDONE en formato JSON	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 63. Caso de prueba Ruta – Enviar Notificación

Caso de prueba	
Número de Caso de Prueba: 17	Número de Historia: 17
Nombre de Caso de Prueba: Ruta – Enviar Notificación	
Descripción: Se debe enviar el Token de autenticación para saber si un usuario es administrador.	
Condiciones de ejecución: Se debe enviar el Token de autenticación, título, descripción, fecha, cédulas, especialidades y enviarlas al servidor en formato JSON	
Entrada/Salida: Token de autenticación, título, descripción, fecha, cédulas, especialidades	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 64. Caso de prueba Ruta – Constancias disponible

Caso de prueba	
Número de Caso de Prueba: 18	Número de Historia: 18
Nombre de Caso de Prueba: Ruta – Constancias disponible	
Descripción:	
Condiciones de ejecución: Se debe enviar el Token de autenticación y la cedula	
Entrada/Salida: Token de autenticación, cedula, tipo de documento y modo de documento en formato JSON	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 65. Caso de prueba Ruta – Generar constancias

Caso de prueba	
Número de Caso de Prueba: 19	Número de Historia: 19
Nombre de Caso de Prueba: Ruta - Generar constancias	
Descripción:	
Condiciones de ejecución: Se debe enviar el Token de autenticación, tipo de documento y modo de documento	
Entrada/Salida: Token de autenticación, tipo de documento y modo de documento en formato, url de documento en formato JSON	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Tabla 66. Caso de prueba App Móvil

Caso de prueba	
Número de Caso de Prueba: 20	Número de Historia: 20
Nombre de Caso de Prueba: App Móvil	

Descripción:
Condiciones de ejecución: Rutas del API REST
Entrada/Salida:
Resultados esperados: No se reportó ningún error de ejecución
Evaluación: Buena

Tabla 67. Caso de prueba Documentación del API REST

Caso de prueba	
Número de Caso de Prueba: 21	Número de Historia: 21
Nombre de Caso de Prueba: Documentación del API REST	
Descripción:	
Condiciones de ejecución: Rutas del API REST	
Entrada/Salida:	
Resultados esperados: No se reportó ningún error de ejecución	
Evaluación: Buena	

Fase IV. Producción

Uno de los aspectos más resaltantes que hay que tener en cuenta al inicio de esta fase es el ambiente de implementación, es decir, las herramientas o entorno en el cual sería desarrollada y ejecutada la aplicación, tomado en consideración elementos del desarrollo tales como: lenguaje de programación, frameworks, entornos de desarrollo integrado,

manejadores de base de datos, herramientas de programación adicionales, entre otros.

Para el desarrollo la web API (Interfaz de Programación de Aplicaciones) con enfoque REST (Transferencia de Estado Representacional) para la gestión de Información Académica Estudiantil, como complemento al Sistema Académico En Línea (SAEL) de la Universidad de Oriente – Núcleo de Nueva Esparta, se empleó el lenguaje de programación PHP, el micro framework FlightPHP para la estructura del REST, JSON para el envío y manipulación de los datos, JSON Web Tokens para la autenticación, el manejador de base de datos Firebird, Android Studio para desarrollar la aplicación móvil, Genymotion y VirtualBox para emular los dispositivos con sistema operativo Android, HTML y CSS3 para la web de la documentación del API REST.

El API REST estuvo a cargo del autor de esta investigación, haciendo uso de las fases y técnicas propuestas en la metodología XP, una vez terminada el API REST, se realizaron pruebas pertinentes y revisiones de rendimiento, pruebas desde la aplicación Android, para verificar la funcionalidad del API.

A continuación, se describen las principales rutas (ver Tabla 68 al 80), algunas interfaces de la aplicación móvil (ver Figuras 3 al 9) y documentación (ver Tabla 10 al 17)

- **Esquema**

Todo el acceso a la API es a través de HTTP, y se accede desde el URL base <http://udone.info/sael-api-restful/api/v1/>. Todos los datos se envían y se reciben en formato de intercambio JSON (JavaScript Object Notation).

Ejemplo: **<http://udone.info/sael-api-restful/api/v1/auth/login>**

- **Agente de usuario**

Define la información del cliente que está accediendo al API, como el nombre de la aplicación, la versión, el sistema operativo, etc.

Ejemplo: **User-Agent:Sael-{client}/{version}**

En el caso de que no se defina un agente de usuario o el cliente de usuario sea web, el token de respuesta para acceder al API tendrá un tiempo de expiración de 600 segundos.

Tabla 68. Definición de User-Agent del API REST

Nombre	Tipo	Ejemplo	Descripción
client	cadena	Mobile, web o null	Tipo de cliente
version	cadena o entero	1.0.0, v1.2.6 o null	Versión del cliente

- **Token de seguridad**

Para acceder a todas las funciones del API (excepto la función de inicio de sesión) se necesita de un token, el cual es generado cuando se inicia sesión con cedula/contraseña o token estáticos para algunas rutas especiales. A partir de entonces, cada petición HTTP que haga el usuario va acompañada de un token en la cabecera. El formato de token está compuesto por 3 cadenas separado por un punto o un token de una 1 cadena.

Ejemplo: **eyJslR5cpXVCJ9.eyJzdWliOil1NY2.yk4nouUtHbWtgg1w**

Ejemplo: **eyJslR5cpXVCJ9**

- **Formato de la cabecera HTTP.**

Ejemplo: **Authorization: Bearer eyJslR5cpXVCJ9.....**

- **Permisos**

Existen dos permisos importantes para los usuarios, estos permisos tienen que estar habilitados para poder acceder al API desde cualquier cliente.

1. Cuando el usuario no tiene una cuenta creada o la cuenta no está activada, dará como resultado una respuesta *403 Forbidden*.
2. Cuando el usuario no ha habilitado la opción para poder utilizar el cliente móvil, dará como resultado una respuesta *403 Forbidden*.

- **Errores de cliente**

1. Envío de JSON no válido o no se envía Content-Type: application/json en la cabecera, dará como resultado una respuesta *400 Bad Request*.
2. Cuando las credenciales son invalidas cedula/contraseña, dará como resultado una respuesta *401 Unauthorized*.
3. Cuando el token ha expirado, dará como resultado *401 Unauthorized*.
4. Cuando no se envía el token en la cabecera, dará como resultado *401 Unauthorized*.
5. Cuando se envía el Bearer pero no el token en la cabecera, dará como resultado *401 Unauthorized*.
6. Cuando se envía el Token, pero no el Bearer en la cabecera, dará como resultado *401 Unauthorized*.
7. Si el Número de los segmentos del Token es incorrecto, dará como resultado *401 Unauthorized*.

8. Cuando el token es invalido, dará como resultado *401 Unauthorized*.
9. Cuando se accede a una ruta que no está permitida para los usuarios, dará como resultado *403 Forbidden*.
10. Cuando se accede a una ruta que no existe, dará como resultado *404 Not Found*.
11. Cuando se envían los datos en el JSON de forma inválida, valores nulos, vacíos o cuando se espera un entero y se envía una cadena, un booleano, etc., dará como resultado *422 Unprocessable Entity*.

- **Errores de servidor**

Errores que puede presentar el API, se tiene que comunicar con los administradores de SAEL para poder corregir el error sucedido y dar soluciones rápidas. Dara como resultado *500 Internal Server Error*.

Rutas del API REST

Tabla 69. API REST Iniciar sesión

Título: Iniciar sesión	URI: auth/login	Método: POST
URL Parámetros:	Datos de Parámetros: <ul style="list-style-type: none"> • cedula: Numéricos • password: Alfanuméricos 	
Código de respuesta: <ul style="list-style-type: none"> • 200: Ok • 401: Unauthorized 	Cabecera: User-Agent:Sael-{client}/{version}	
Respuesta: <pre>{ "token": "eyJ0eXkiJ9.eyJjZWVhbnR0.XUk1Dyl", "expires": "not expire" }</pre>		

Tabla 70. API REST Perfil

Título: Perfil	URI: auth/me	Método: GET
URL Parámetros:	Datos de Parámetros:	
Código de respuesta: <ul style="list-style-type: none"> • 200: Ok • 401: Unauthorized 	Cabecera: Authorization: Bearer eyJslnR5cpXVCJ9....	
Respuesta: <pre>{ "ID": "3", "CEDULA": "21015369", "NOMBRE1": "Roger", }</pre>		

```

"NOMBRE2": "Aldayl",
"APPELLIDO1": "Patiño",
"APPELLIDO2": "Blanco",
"CORREO": "rogerp.009@gmail.com",
"ANO_ING": "2010",
"TITULO": "LIC. INFORMATICA",
"CODIGO": "12983",
"PROMEDIO": "6.175000"
}

```

Tabla 71. API REST Medidas

Título: Medidas	URI: measures	Método: GET
URL Parámetros:	Datos de Parámetros:	
Código de respuesta: <ul style="list-style-type: none"> • 200: Ok • 404: Not Found 	Cabecera: <p>Authorization: Bearer eyJslnR5cpXVCJ9....</p>	
Respuesta: <pre> [{ "ANO": "2013", "PER": "1", "CEDULA": "20111253", "MEDIDA": "25" }] </pre>		

Tabla 72. API REST Notas

Título: Notas	URI: notes	Método: GET
URL Parámetros:	Datos de Parámetros:	
Código de respuesta: <ul style="list-style-type: none"> • 200: Ok • 404: Not Found 	Cabecera: <p>Authorization: Bearer eyJslnR5cpXVCJ9....</p>	
Respuesta: <pre> [</pre>		

```
{
  "NOMBRE": "MATEMATICAS II",
  "CODIGO": "0081224",
  "ULT_ANO": "2010",
  "ULT_PER": "3",
  "ULT_NOTA": "09",
  "INSCRITA": "1",
  "RETIRADA": "0",
  "REPROBADA": "0",
  "NOREPORTADA": "0",
  "APROBADA": "1"
},
]
```

Tabla 73. API REST Apelación

Título: Apelación	URI: appeals	Método: GET
URL Parámetros:		Datos de Parámetros:
Código de respuesta: <ul style="list-style-type: none"> • 200: Ok • 404: Not Found 		Cabecera: Authorization: Bearer eyJslnR5cpXVCJ9....
Respuesta: <pre>[{ "ANO": "2012", "PER": "1", "CEDULA": "22012012", "CREADO": "2014-10-23 12:03:21", "MOTIVO": "1", "ENTREGADO": "0", "APROBADO": "0", "COMENTARIO": "NO PROCEDE" }]</pre>		

Tabla 74. API REST Escuelas

Título: Escuelas	URI: schools	Método: GET
------------------	--------------	-------------

URL Parámetros:	Datos de Parámetros:
Código de respuesta: <ul style="list-style-type: none"> • 200: Ok • 404: Not Found 	Cabecera: <p>Authorization: Bearer eyJslnR5cpXVCJ9....</p>
Respuesta: <pre>[{ "CODIGO": "3", "DESCRIPCION": "CIENCIAS APLICADAS DEL MAR", "ESTATUS": "1" }]</pre>	

Tabla 75. API REST Especialidad

Título: Especialidad	URI: specialtys	Método: GET
URL Parámetros:	Datos de Parámetros:	
Código de respuesta: <ul style="list-style-type: none"> • 200: Ok • 404: Not Found 	Cabecera: <p>Authorization: Bearer eyJslnR5cpXVCJ9....</p>	
Respuesta: <pre>[{ "CODIGO": "18592", "TITULO": "LIC. ADMINISTRACION", "PENSUM": "1" }]</pre>		

Tabla 76. API REST Pensum

Título: Pensum	URI: <ul style="list-style-type: none"> • specialty/:id_specialty/pensum/:id_pensum 	Método: GET
----------------	--	-------------

<p>URL Parámetros:</p> <ul style="list-style-type: none"> id_specialty: Cadena, Identificador de la especialidad id_pensum: Entero, Identificador del pensum (0 o 1), esto dependerá del año, puede ser pensum viejo o nuevo 	<p>Datos de Parámetros:</p>
<p>Código de respuesta:</p> <ul style="list-style-type: none"> 200: Ok 404: Not Found 	<p>Cabecera:</p> <p>Authorization: Bearer eyJslnR5cpXVCJ9....</p>
<p>Respuesta:</p> <pre>[{ "CARRERA": "18592", "PENSUM": "1", "CODIGO": "0061922", "NOMBRE": "EXPRESION ORAL Y ESCRITA", "TIPO_MATERIA": "PO", "TIPO_MATERIA_NOMBRE": "PROFESIONAL OBLIGATORIA", "SEMESTRE": "01", "CREDITOS": "2", "ESPECIAL": "0", "REQUISITOS": [] }]</pre>	

Tabla 77. API REST Materia

<p>Título: Materia</p>	<p>URI:</p> <ul style="list-style-type: none"> specialty/:id_specialty/pensum/:id_pensum/matter/:id_matter 	<p>Método: GET</p>
<p>URL Parámetros:</p>	<p>Datos de Parámetros:</p>	
<p>Código de respuesta:</p> <ul style="list-style-type: none"> 200: Ok 404: Not Found 	<p>Cabecera:</p> <p>Authorization: Bearer eyJslnR5cpXVCJ9....</p>	
<p>Respuesta:</p> <pre>{ "TITULO": "LIC. ADMINISTRACION", "ESPECIALIDAD_CODIGO": "18592", "PENSUM": "1", "NOMBRE": "EXPRESION ORAL Y ESCRITA", "SEMESTRE": "01", "CREDITOS": "2", "CODIGO_MATERIA": "0061922", "TIPO": "PO", "TIPO_MATERIA": "PROFESIONAL OBLIGATORIA", "REQUISITOS": [] }</pre>		

Tabla 78. API REST Notificación

Título: Notificación	URI: notification	Método: GET
URL Parámetros:	Datos de Parámetros: <ul style="list-style-type: none"> • title: cadena • body: cadena • time: cadena • from_to: cadena • cedula: cadena • carrerar: cadena 	
Código de respuesta: <ul style="list-style-type: none"> • 200: Ok • 404: Not Found 	Cabecera: Authorization: Bearer eyJslnR5cpXVCJ9....	
Respuesta: <pre>{ "TITULO": "LIC. ADMINISTRACION", "ESPECIALIDAD_CODIGO": "18592", "PENSUM": "1", "NOMBRE": "EXPRESION ORAL Y ESCRITA", "SEMESTRE": "01", "CREDITOS": "2", "CODIGO_MATERIA": "0061922", "TIPO": "PO", "TIPO_MATERIA": "PROFESIONAL OBLIGATORIA", "REQUISITOS": [] }</pre>		

Tabla 79. API REST Chechar Documento

Título: Chechar Documento	URI: doc_check	Método: GET
URL Parámetros:	Datos de Parámetros: <ul style="list-style-type: none"> • cedula: Numérico 	
Código de respuesta: <ul style="list-style-type: none"> • 200: Ok • 404: Not Found 	Cabecera: <ul style="list-style-type: none"> • Authorization: Bearer eyJslnR5cpXVCJ9.... • User-Agent Sael-Mobile/1.x.x • Accept application/json 	

```

Respuesta:
{
"doc_modo":{ "i":"Interno","e":"Externo (requiere Pago y Firma del Jefe)",
"f":"Beca Funjemasu (solicitud o renovacion)","fo":"Fontur-NE (solicitud de tarjeta o
recarga)"
},
"doc_tipo":{
"ci":"Constancia de Inscripcion", "ce":"Constancia de Estudios","re":"Record Academico"
}
}

```

Tabla 80. API REST Generar Documento

Título: Generar Documento	URI: doc_generate	Método: GET
URL Parámetros: <ul style="list-style-type: none"> cedula: Numérico 	Datos de Parámetros: <ul style="list-style-type: none"> cedula: numérico, doc_modo: carácter, ci: numérico ce: numérico re: numérico 	
Código de respuesta: <ul style="list-style-type: none"> 200: Ok 404: Not Found 	Cabecera: <ul style="list-style-type: none"> Authorization: Bearer eyJslnR5cpXVCJ9.... User-Agent Sael-Mobile/1.x.x 	
Respuesta: <pre> { "code": "200", "doc_id": 63920, "link": "http://udone.info/cache/mobile_docs/7c2c84f157e75705e07da5e8b8e25447.pdf" } </pre>		

Aplicación Móvil usando las Rutas del API REST



Figura 2. Iniciar sesión de aplicación móvil

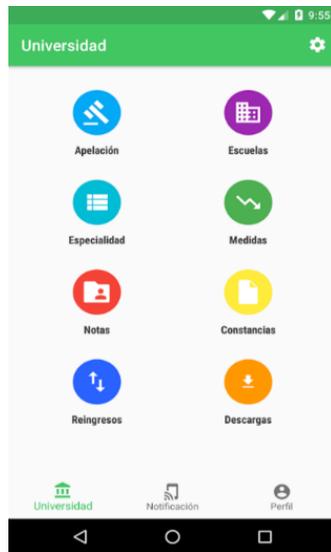


Figura 3. Menú con opciones de aplicación móvil



Figura 4. Notificación de aplicación móvil

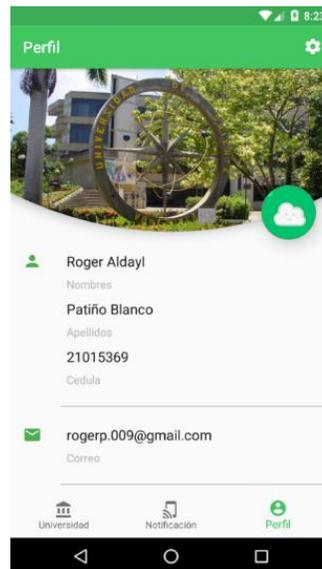


Figura 5. Perfil de aplicación móvil

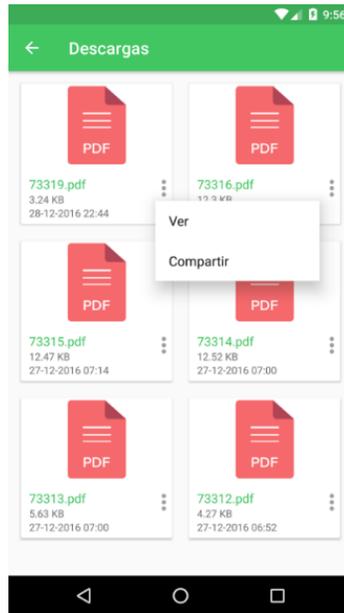


Figura 6. Documentos de aplicación móvil



Figura 7. Especialidad de aplicación móvil

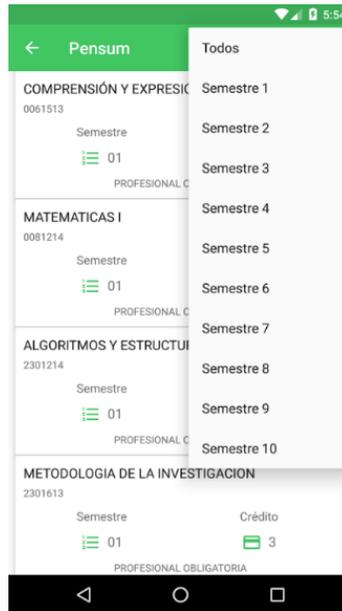


Figura 8. Pensum de aplicación móvil

Documentación de las Rutas de API REST

UDONE SAEL - Developer

v1 (API REST)

- AUTORIZACIÓN
 - Iniciar sesión
 - Perfil
 - Iniciar sesión y Perfil
 - Carnet
- APELACIONES
 - Apelación
- ESCUELA
 - Escuelas
 - Escuela
- ESPECIALIDAD
 - Especialidades
 - Especialidad
 - Pensum
 - Materia
 - Semestre
 - Electivas
 - Tipo materia

Documentación API

Esto describe los recursos que componen la versión 1 del API oficial Sael.

Esquema

Todo el acceso a la API es a través de HTTP, y se accede desde el `http://udone.info/sael/api/v1/..` Todos los datos se envían y se reciben como formato de intercambio JSON (JavaScript Object Notation).

Ejemplo: `http://udone.info/sael/api/v1/auth/login`

Agente de usuario

Definir la información del cliente que está accediendo a las API, como el nombre de la aplicación, la versión, el sistema operativo, etc.

Ejemplo: `User-Agent:Sael-{client}/{version}`

En el caso de que no se defina un agente de usuario o está sea web, el token de respuesta para acceder las API, tendrá un tiempo de expiración de 600 segundos.

Nombre	Tipo	Ejemplo	Descripción

Figura 9. Esquemas y estructura de enviar parámetros en rutas del API REST

UDONE SAEL - Developer

v1 (API REST)

AUTORIZACIÓN
 Iniciar sesión
 Perfil
 Iniciar sesión y Perfil
 Carnet

APELACIONES
 Apelacion

ESCUELA
 Escuelas
 Escuela

ESPECIALIDAD
 Especialidades
 Especialidad
 Pensum
 Materia
 Semestre
 Electivas
 Tipo materia

Documentación API

Esto describe los recursos que componen la versión 1 del API oficial Sael.

Iniciar sesión

Ruta

POST `auth/login`

Cuerpo

Nombre	Tipo	Descripción
<code>cedula</code>	String	Cedula del usuario
<code>password</code>	String	Contraseña del usuario

```

{
  "cedula": "21015369",
  "password": "3utUY6"
}

```

Figura 10. Iniciar sesión en las rutas del API REST

Iniciar sesión y Perfil
 Carnet

APELACIONES
 Apelacion

ESCUELA
 Escuelas
 Escuela

ESPECIALIDAD
 Especialidades
 Especialidad
 Pensum
 Materia
 Semestre
 Electivas
 Tipo materia

MEDIDAS
 Medida

NOTAS
 Notas

Perfil

Ruta

GET `auth/me`

Respuestas

Estado: 200 Ok

```

{
  "ID": "3",
  "CEDULA": "21015369",
  "NOMBRE1": "Rogen",
  "NOMBRE2": "Aldayl",
  "APELLIDO1": "Patiño",
  "APELLIDO2": "Blanco",
  "CORREO": "rogerp.009@gmail.com",
  "CARRERA2": null,
  "ANO_ING": "2010",
  "ESTATUS": "1",
  "TELF": null,
  "DIRECCION": null,
  "TITULO": "LIC. INFORMATICA",
  "CODIGO": "12983",
  "PROMEDIO": "6.175000"
}

```

Figura 11. Perfil en las rutas del API REST

Todas las escuelas

Ruta

GET schools

Respuestas

Estado: 200 Ok

```
[
  {
    "CODIGO": "1",
    "DESCRIPCION": "UNIDAD DE ESTUDIOS BÁSICOS",
    "ESTATUS": "1"
  },
  {
    "CODIGO": "2",
    "DESCRIPCION": "HOTELERÍA Y TURISMO",
    "ESTATUS": "1"
  },
  {
    "CODIGO": "3",
    "DESCRIPCION": "CIENCIAS APLICADAS DEL MAR",
    "ESTATUS": "1"
  }
]
```

Figura 12. Escuelas en las rutas del API REST

Apelación

Ruta

GET appeals

Respuestas

Estado: 200 Ok

```
{
  "ANO": "2012",
  "PER": "1",
  "CEDULA": "22012012",
  "CREADO": "2014-10-23 12:03:21",
  "MODIFICADO": null,
  "TIPO": "1",
  "SUBTIPO": "0",
  "REPOSO": "0",
  "NOMBRE": null,
  "TELF": null,
  "ANEXOS": null,
  "MOTIVO": "1",
  "ENTREGADO": "0",
  "APROBADO": "0",
  "COMENTARIO": "NO PROCEDE",
}
```

file:///C:/Users/Roger/Documents/Teis/rogerp91-docsapi-rest-d1e23c21ba1d/index.html#appeals

Figura 13. Apelación en las rutas del API REST

Todas las especialidades

Ruta

GET /especialty

Respuestas

Estado: 200 Ok

```
[
  {
    "CODIGO": "13017",
    "TITULO": "LIC. ACUACULTURA",
    "PENSUM": "0"
  },
  {
    "CODIGO": "18592",
    "TITULO": "LIC. ADMINISTRACION",
    "PENSUM": "1"
  }
]
```

Estado: 200 Ok

Figura 14. Especialidad en las rutas del API REST

Pensum de especialidad

Ruta

GET /specialty/:id_specialty/pensum/:id_pensum

Parametro

Nombre	Tipo	Descripción
id_specialty	String	Identificador de la especialidad
id_pensum	Int	Identificador del pensum (0 o 1), esto dependerá del año, puede ser pensum viejo o nuevo. Normalmente los pensum nuevos son identificado por el número 1.

Respuestas

Estado: 200 Ok

```
[
  {
    "CARRERA": "18592",
    "PENSUM": "1",
    "CODIGO": "0061922",
    "NOMBRE": "EXPRESSION ORAL Y ESCRITA",
    "TIPO_MATERIA": "PO",
    "TIPO_MATERIA_NOMBRE": "PROFESIONAL OBLIGATORIA",
  }
]
```

file:///C:/Users/Roger/Documents/Tesis/rogerp91-docsairest-d1e23c21ba1d/index.html#/pensum

Figura 15. Pensum en las rutas del API REST

Iniciar sesión y Perfil
Carnet

APELACIONES
Apelación

ESCUELA
Escuelas
Escuela

ESPECIALIDAD
Especialidades
Especialidad
Pensum
Materia
Semestre
Electivas
Tipo materia

MEDIDAS
Medida

NOTAS
Notas

PROFESOR

Reingreso

Ruta

GET readmissions

Respuestas

Estado: 200 Ok

```
[
  {
    "ANO": "2012",
    "PER": "3",
    "CEDULA": "20456998",
    "CREADO": "2014-10-23 12:00:18",
    "OFICIO": "1",
    "FECHA_OFICIO": null,
    "CARRERA_NUEVA": null,
    "COMENTARIO": null,
    "EGRESADO": "0",
    "APROBADO": "1"
  },
  {
    "ANO": "2014",
    "PER": "1",
    "CEDULA": "20456998",
    "CREADO": "2014-10-23 12:00:19",
    "OFICIO": "1"
  }
]
```

Figura 16. Reingreso en las rutas del API REST

CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

La investigación desarrollada Web API (Interfaz de programación de aplicaciones) con enfoque REST (Transferencia de Estado Representacional) para la gestión de Información Académica Estudiantil, como complemento al Sistema Académico En Línea (SAEL) de la Universidad de Oriente – Núcleo de Nueva Esparta, fue para dar solución a la necesidad planteada a través de la observación directa de las tareas y actividades.

En forma precisa y metodológica, se planteó el objetivo general y los objetivos específicos, los cuales se cumplieron a cabalidad, permitiendo llegar a las siguientes conclusiones:

1. Se analizó bases teóricas necesarias para el diseño y la estructura de una web API REST, estudio de los verbos y términos de métodos HTTP, procesamiento de solicitudes HTTP, tipos y formatos de respuesta HTTP.
2. Durante la fase de Exploración de la metodología XP, se dieron a conocer cuáles fueron los requerimientos basado en sistema actual SAEL que conllevan a la creación del API REST, entre ellos: Inicio de sesión, medidas, apelaciones escuelas, pensum, electivas, notas, descarga de documentos y notificaciones.

3. Aplicando las etapas de la metodología XP, se diseñaron 19 rutas nuevas, que corresponden al API REST para la gestión de Información Académica Estudiantil como complemento al sistema SAEL, las cuales se basaron en las Historias de Usuario, y fueron hechas con el fin de dar respuesta a los requerimientos exigidos.
4. Se programaron las rutas del API REST el lenguaje programación PHP, el framework FlightPHP para la estructura del REST, JSON para el envío y manipulación de los datos, JSON Web Tokens para la autenticación, el manejador de base de datos Firebird, las rutas se crearon siguiendo la arquitectura REST, basándose en las siguientes premisas: cliente/servidor, cache, sin estado y uniformes, esto significa que puede ser manipulado por varios cliente al mismo tiempo.
5. Para el funcionamiento de las rutas del API REST se implementó un prototipo utilizando las tecnologías y framework del sistema operativo Android, en un entorno del lenguaje Java para la aplicación móvil. se utilizaron los programas Genymotion y VirtualBox para emular los dispositivos con sistema operativo Android.
6. Se utilizaron las tecnologías web HTML y CSS3 para la creación de una página web simple que sirve para establecer una documentación y mostrar el funcionamiento de las rutas del API REST.
7. Se realizaron pruebas pertinentes y se verificó el correcto funcionamiento de las rutas en la aplicación móvil.

RECOMENDACIONES

Se desea que esta investigación, sirva como bases a otras futuras, es por ello que se recomienda a las personas interesados en este tema, seguir las siguientes recomendaciones:

- ✓ Evaluar durante el periodo de uso, si se presenta alguna falla para aplicar la quinta fase (Mantenimiento) y sexta fase (Muerte del proyecto) de la metodología XP, si fuese necesario.
- ✓ Respetar y mantener los estándares en cuanto al diseño del API REST, al momento de añadir otros módulos, en caso de que éste así lo amerite.
- ✓ La creación de un nuevo prototipo para el sistema operativo iOS de Apple.
- ✓ Para probar el prototipo Android, se recomienda el uso de dispositivos con sistema operativo Android superior a 4.1 Jelly Bean.
- ✓ Supervisar y evaluar las rutas del API REST y del prototipo una vez que los usuarios comiencen a hacer uso del mismo, mediante el rastreo de datos, esto con el fin de comprobar su correcto uso y funcionamiento.

- ✓ Se recomienda la implementación en el sistema educativo de metodologías ágiles para el desarrollo de software como: Kanban, Programación Extrema (XP) y Scrum.
- ✓ Se recomienda incentivar a los estudiantes sobre el estudio y desarrollo de las tecnologías móviles.

REFERENCIAS

ABC Tecnología, (2015). ¿Qué es una API y para qué sirve? [Página Web en Línea], Disponible: <http://goo.gl/LPzoR0> [Consulta: 2016, Febrero 25].

Álvarez, M. (2001). Qué es la programación orientada a objetos. [Página Web en Línea]. Disponible en: <https://goo.gl/Q1hFkE> [Consulta: 2016, Octubre 07].

Arce, M. (2012). Desarrollo e implementación de una API RESTful aplicada a un Sistema Académico Universitario [Tesis en Línea] Universidad de Mendoza. Argentina. Consultada el 25 de febrero de 2016 en: <http://goo.gl/GZuEy6>.

Área Tecnología, (2010). ¿Qué es un Smartphone? [Página Web en Línea]. Disponible en: <https://goo.gl/EDcE91> [Consulta: 2016, Octubre 05].

Arias, F. (1999). El Proyecto de Investigación. (3ra ed.). Caracas, Venezuela: Editorial Episteme.

Bavaresco, A. (2006) Proceso metodológico en la investigación (Cómo hacer un Diseño de Investigación). Maracaibo, Venezuela: Editorial de la Universidad del Zulia.

BBVAOpen4U. (2016). API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos. [Página Web en Línea]. Disponible en: <https://goo.gl/0NboeA> [Consulta: 2016, Octubre 07].

Beck, K. (2000). Extreme Programming Explained, Boston, Estados Unidos: Editorial Addison-Wesley.

Bringas, A. (2000). ¿Qué es Firebird? [Página Web en Línea], Disponible: <https://goo.gl/turG8p> [Consulta: 2016, Septiembre 20].

Bustamante, J. (2011). Introducción al Diseño Web Adaptable o Responsive Web Design. [Documento en línea], Disponible: <http://goo.gl/hRFQEW> [Consulta: 2016, Marzo 19].

Cambara, J. (2012). Composición de Servicios Web para Aplicaciones Móviles Geolocalizadas [Tesis en Línea] Universidad Simón Bolívar. Venezuela. Consultada el 25 de febrero de 2016 en: <http://goo.gl/Hs3k55>.

Chacón, A y Teodoro, L. (2010). Diseño De Una Aplicación Móvil Para La Consulta Académica De La Facultad de Ingeniería Industrial y de Sistemas (FIIS). [Tesis en Línea]. Universidad Tecnológica del Perú. Perú. Consultada el 22 de Agosto de 2016 en: <https://goo.gl/iauaBD>.

Cruz, J. (2011). Tecnologías Móviles. [Página Web en Línea]. Disponible en: <https://goo.gl/jA8DtR> [Consulta: 2016, Octubre 12].

Cockbun, A y Williams, L. The Costs and Benefits of Pair Programming. Enero, 2000. Disponible: <https://goo.gl/juxPk0>. [Consulta: 2016, Octubre 05].

Comisión Federal de Comercio. (2011). Aplicaciones móviles: Qué son y cómo funcionan. [Página Web en Línea]. Disponible en: <https://goo.gl/A5DZdf> [Consulta: 2016, Octubre 10].

Corrales, J y Arribas, M. (2006) Conocimientos básicos de informática. Primera edición. Editorial Mad, S.L. Sevilla, España.

Díaz, M. (2011). Desarrollo de un sistema automatizado basado en inteligencia de negocio, que integre los procesos administrativos del almacén, del súper mercado Bello Monte. [Tesis en Línea]. Universidad Nueva Esparta. Venezuela. Consultada el 20 de Marzo de 2016 en: <http://goo.gl/B7uq3H>.

EcuRed. (2009). Servidor Web. [Página Web en Línea]. Disponible en: <https://goo.gl/PNxAAm> [Consulta: 2016, Octubre 15].

ECMA-262 (ISO/IEC 16262), ECMAScript® Language Specification, 3rd edition (December 1999).

Fernández, O. (2004). Introducción al lenguaje de programación Java. [Página Web en Línea]. Disponible en: <https://goo.gl/UGGr9Q> [Consulta: 2016, Octubre 10].

Fowler, M, Beck, K, Brant, J. (1999). Refactoring: Improving the Design of Existing Code. USA: Editorial: Addison-Wesley.

Gutiérrez, J. (2014). ¿Qué es un framework web? [Página Web en Línea]. Disponible en: <https://goo.gl/ftcY6U> [Consulta: 2016, Octubre 05].

Hernández, H. (2010). Tecnología móvil. [Página Web en Línea]. Disponible en: <https://goo.gl/u616Au> [Consulta: 2016, Octubre 05].

Hurtado, J. (2000). Metodología de la investigación holística. (3ra. ed.). Caracas, Venezuela: Editorial Sypal.

ISPAMAT. (2017). Aplicación móvil: ¿Web o nativa? [Página Web en Línea]. Disponible en: <https://goo.gl/rLgbYz> [Consulta: 2016, Octubre 10].

Joyanes, L y Fernández, M. (2001). Java 2 Manual de Programación. 1ra Edición, Editorial Mc Graw Hill.

Laboratorio Nacional de Calidad de Software. (2009). Ingeniería del Software: Metodologías y Ciclos de Vida. INTECO. España.

Lázaro, D. (2016). Códigos de estado en las respuestas HTTP que un servidor puede devolver cuando recibe una petición HTTP request. [Página Web en Línea]. Disponible en: <https://goo.gl/Fq6Jn5> [Consulta: 2016, Octubre 05].

Louden, K. (2002) Lenguajes de Programación: Principios y Prácticas. Cengage Learning Editores. México.

Mancera, R. (2013). Web Responsiva: diseña tu web para cualquier dispositivo. [Documento en línea], Disponible: <http://goo.gl/khXFYk> [Consulta: 2016, Marzo 19].

Mi BQ y Yo, (2015). Aprende sobre las notificaciones Push. [Página Web en Línea], Disponible: <https://goo.gl/65i2fl> [Consulta: 2016, Septiembre 20].

Navarro, R. (2016). REST vs Web Server. [Documento en línea], Disponible: <http://goo.gl/vNdXmM> [Consulta: 2016, Febrero 25].

OboLog, (2012). METODOLOGIA XP. [Página Web en Línea], Disponible: <https://goo.gl/pDTnS4> [Consulta: 2016, Febrero 28].

OCU Ediciones, S.A (2011). Sistema operativo para móviles. [Página Web en Línea]. Disponible en: <https://goo.gl/lk9hnm> [Consulta: 2016, Octubre 10].

Open Handset Alliance. (2007). Android. [Página Web en Línea]. Disponible en: <https://goo.gl/uXs1GG> [Consulta: 2016, Octubre 05].

Pumarino, J. (2010). Desarrollo De Plugins Distribuidos Como Servicios Web REST [Tesis en Línea]. Universidad Católica de Chile. Chile. Consultada el 25 de febrero de 2016 en: <http://goo.gl/tZoq1z>.

Revelo, J. (2016). Android Push Notifications Con Firebase Cloud Messaging. [Página Web en Línea], Disponible: <https://goo.gl/wPu87z> [Consulta: 2016, Septiembre 20].

Rivera, G. (2002). Diseño e implementación de un Web API para el Sistema Interactivo de Desarrollo para el Web (SIDWeb) [Tesis en Línea]. Escuela Superior Politécnica Del Litoral. Ecuador. Consultada el 25 de febrero de 2016 en: <http://goo.gl/oaDuju>.

Sabino, C. (1997). Metodología de la investigación, Caracas, Venezuela: Editorial Panapo.

Soto, M. (2014). Desarrollo De Una Aplicación Móvil Android Para Mejorar La Integración De Los Estudiantes De Intercambio En La UPV Mediante Uso De Herramientas Útiles. [Tesis en Línea]. Escuela Técnica Superior de ingeniería informática Universidad Politécnica de Valencia. España. Consultada el 12 de Octubre de 2016 en: <https://goo.gl/6qslg7>.

Tamayo y Tamayo, M. (1987). El proceso de la Investigación Científica. (2da. Ed.) México: Editorial Limusa.

Tamplin, T. (2016). Firebase se expande para convertirse en una plataforma de apps unificada. [Página Web en Línea]. Disponible en: <https://goo.gl/nNmv6j> [Consulta: 2016, Octubre 07].

Universidad Carlos III de Madrid. (2011). “Programación en dispositivos móviles portables. Android”. [Página Web en Línea]. Disponible en: <https://goo.gl/EKYNlz> [Consulta: 2016, Octubre 05].

Universidad Pedagógica Experimental Libertador (UPEL). (1998). Manual de Trabajos de Grado de especialización y Maestría y Tesis Doctorales. (2da. Ed.) Caracas: FEDUPEL.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

TÍTULO	Desarrollo de una web API (Interfaz de programación de aplicaciones) con enfoque REST (Transferencia de Estado Representacional) para la gestión de Información Académica Estudiantil, como complemento al Sistema Académico En Línea (SAEL) de la Universidad de Oriente – Núcleo de Nueva Esparta.
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
Patiño Blanco, Roger Aldayl	CVLAC: 21015369 E MAIL: rogerp.009@gmail.com
	CVLAC: E MAIL:

	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:

- **UDONE**
- **SAEL**
- **API**
- **REST**
- **ANDROID**
- **XP**
- **DOC**

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ÀREA	SUBÀREA
Ingeniería y Ciencias Aplicadas	Informática.

RESUMEN (ABSTRACT): Actualmente, la Universidad de Oriente, no cuenta con un servicio web API REST o con algún otro mecanismo de desarrollo para obtener la información académica y estudiantil por medio de un cliente web o móvil. Es por ello, que la presente investigación, se basa en la creación de un servicio web API con arquitectura REST para obtener la data de la universidad, como complemento al sistema académico SAEL. Se utilizó la metodología ágil Extreme Programming (XP) de Kent Beck, también se utilizó el lenguaje programación PHP,

Bases de Datos Firebird y para comprobar la operatividad de servicio Web, se implementó una aplicación móvil como prototipo y así presentar una nueva alternativa para realizar las diversas consultas académicas y la administración de información referente a la universidad como publicaciones, eventos, inscripciones o noticias mediante notificaciones push.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

CONTRIBUIDORES:

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
Lcdo.: Arabia Alfredo.	ROL	CA	AS	TU X	JU X
	CVLAC:	9.424.163			
	E_MAIL	alfredo.arabia@gmail.com			
	E_MAIL				
Lcdo.: José Ordaz.	ROL	CA	AS	TU	JU
	CVLAC:				
	E_MAIL	josearcadioordaz@gmail.com			
	E_MAIL				
MSc.: Margarita Fajardo.	ROL	CA	AS	TU	JU
	CVLAC:	12.618.114			
	E_MAIL				
	E_MAIL				
	ROL	CA	AS x	TU	JU

	CVLAC:	
	E_MAIL	
	E_MAIL	

FECHA DE DISCUSIÓN Y APROBACIÓN:

2016	Junio	02
AÑO	MES	DÍA

LENGUAJE. SPA

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ARCHIVO (S):

NOMBRE DE ARCHIVO	TIPO MIME
Trabajo_de_grado_Patino_Roger_Blanco_Aldayl	.docx
Trabajo_de_grado_Patino_Roger_Blanco_Aldayl	.pdf

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F G H
I J K L M N O P Q R S T U V W X Y Z . a b c d e f g h i j k l m n o p q r s t u
v w x y z . 0 1 2 3 4 5 6 7 8 9 .

ALCANCE

ESPACIAL: _____ **(OPCIONAL)**

TEMPORAL: _____ **(OPCIONAL)**

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

Licenciatura en Informática.

NIVEL ASOCIADO CON EL TRABAJO:

Pregrado.

ÁREA DE ESTUDIO:

Informática.

**INSTITUCIÓN: Universidad de Oriente, Núcleo Nueva
Esparta**

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

DERECHOS

Artículo 41 del reglamento de trabajo de pregrado (Vigente a partir del II semestre 2009, Según comunicado CU-034-2009). "Los Trabajos de Grado son Propiedad exclusiva de la Universidad y sólo podrán ser utilizados para otros fines con el consentimiento del Consejo de Núcleo respectivo, quien lo participará en Consejo Universitario"

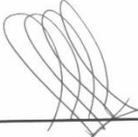


Roger Patiño

AUTOR



Alfredo Arabia
TUTOR



José Ordaz
JURADO



Margarita Fajardo
JURADO

POR LA COMISION DE TRABAJO DE GRADO