

CIS1410IS05
ARQUITECTURA ORIENTADA A RECURSOS PARA APLICACIONES
HIPERMEDIALES EN CONTEXTOS E-LEARNING.

ANGELICA MARIA VERGARA GRANADOS

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
CARRERA DE INGENIERIA DE SISTEMAS
BOGOTÁ, D.C.
2014

CIS1410IS05

Arquitectura Orientada a Recursos Para Aplicaciones Hipermediales En contextos E-learning

Autor(es):

Angélica María Vergara Granados

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO
DE LOS REQUISITOS PARA OPTAR AL TITULO DE INGENIERO DE
SISTEMAS

Director

Julio Ernesto Carreño Vargas

Jurados del Trabajo de Grado

<Nombres y Apellidos Completos del Jurado >

<Nombres y Apellidos Completos del Jurado >

Página web del Trabajo de Grado

<http://pegasus.javeriana.edu.co/~CIS1410IS05/>

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
CARRERA DE INGENIERIA DE SISTEMAS
BOGOTÁ, D.C.
Junio, 2014

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
CARRERA DE INGENIERIA DE SISTEMAS**

Rector Magnífico

Jorge Humberto Peláez Piedrahita S.J.

Decano Académico Facultad de Ingeniería

Ingeniero Jorge Luis Sánchez Téllez

Decano del Medio Universitario Facultad de Ingeniería

Padre Antonio José Sarmiento Nova S.J.

Director de la Carrera de Ingeniería de Sistemas

Ingeniero Germán Alberto Chavarro Flórez

Director Departamento de Ingeniería de Sistemas

Ingeniero Rafael Andrés González Rivera

Artículo 23 de la Resolución No. 1 de Junio de 1946

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”

AGRADECIMIENTOS

Primero que todo quiero dar gracias a Dios por darme la fuerza, paciencia y perseverancia, necesarias para lograr salir adelante cada semestre. A mis padres Carlos Vergara y Josefina Granados, por el apoyo que me brindaron día a día durante mi carrera dándome a conocer que siempre estarán ahí cuando más los necesite. A mi segunda madre, mi tía Nelly Granados, por esos pequeños detalles que siempre ha tenido conmigo, brindándome su apoyo incondicional desde que nací. A mi novio César Quintero por brindarme todo su apoyo y consejos que me ayudaron a ser perseverante siempre. A mis amigos de toda la carrera Juan Murcia, Carlos Corzo y Kevin Vasquez porque gracias a ellos el día a día y la rutina cambiaron constantemente. A mi director de trabajo de grado, Julio Carreño por guiarme en esta ardua labor no sólo como maestro sino como amigo. A Maida Urrego porque fue una amiga más durante toda la carrera y me apoyó cada momento. A ellos y muchas personas más sólo resta decirles gracias por llegar a mi vida y por apoyarme logrando este sueño.

Contenido

INTRODUCCIÓN	10
I - DESCRIPCION GENERAL DEL TRABAJO DE GRADO.....	11
1. OPORTUNIDAD, PROBLEMÁTICA, ANTECEDENTES.....	11
1.1 Descripción del contexto	11
1.2 Formulación del problema que se resolvió	14
1.3 Justificación.....	14
1.4 Impacto Esperado.....	15
2. DESCRIPCIÓN DEL PROYECTO	15
2.1 Visión global.....	15
2.3 Objetivo general	16
2.4 Fases Metodológicas o conjunto de objetivos específicos	16
2.5 Método que se propuso para satisfacer cada fase metodológica.....	17
II - MARCO TEÓRICO	18
1. MARCO CONTEXTUAL.....	18
2. MARCO CONCEPTUAL	24
III – DESARROLLO DEL TRABAJO	39
IV – PROTOTIPO DE APLICATIVO ROA PARA E-LEARNING	77
V – IMPLEMENTACIÓN DEL PROTOTIPO.....	95
VI - RESULTADOS Y REFLEXIÓN SOBRE LOS MISMOS	103
VII – CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS	105
1. CONCLUSIONES	105
2. TRABAJOS FUTUROS	106
VIII - REFERENCIAS Y BIBLIOGRAFÍA.....	107
IX - ANEXOS	109
ANEXO 1. DOCUMENTO SOBRE ARQUITECTURA ORIENTADA A RECURSOS.....	109
ANEXO 2. DOCUMENTO CON OBJETOS VIRTUALES DE APRENDIZAJE EN E-LEARNING ..	109

ANEXO 3. DOCUMENTO CON OBJETOS VIRTUALES DE APRENDIZAJE PARA UNA ARQUITECTURA ORIENTADA A RECURSOS EN E-LEARNING109

ANEXO 4. DOCUMENTO CON PROCESO DE APRENDIZAJE SELECCIONADO109

ANEXO 5. S.A.D (SOFTWARE ARCHITECTURE DOCUMENT)109

ANEXO 6. DOCUMENTO CON OBJETOS VIRTUALES DE APRENDIZAJE CREADOS Y SU ESPECIFICACIÓN CORRESPONDIENTE109

ANEXO 7. DOCUMENTACIÓN DEL PROTOTIPO.....109

ANEXO 8. CREACIÓN DE UN CURSO E-LEARNING (CASO DE ESTUDIO).....109

ANEXO 9. GLOSARIO.....109

ABSTRACT

The main objective of this work is build a Resource Oriented Architecture supporting an application protocol (business process) using hipermedia as a principle between REST services and the Architecture mentioned at the begining, whitin e-learning context. Giving and answer to what was mentioned before, it begins giving an explanation about the creation of a e-learning course with its principal elements, then is defined a learning process in e-learning context which will be the application protocol. Finally, it start defining an Resource Oriented Architecture, based on the learning process.

RESUMEN

El objetivo principal de este trabajo de grado es construir una arquitectura orientada a recursos soportando un protocolo de aplicación (proceso de negocio), haciendo uso de la hipermedia como principio fundamental entre los servicios REST y la arquitectura ya mencionada, todo enfocado al contexto de e-learning. Dando respuesta a lo mencionado anteriormente, se comienza explicando la creación de un curso e-learning con sus principales elementos, posteriormente se define el proceso de aprendizaje adecuado para e-learning que será tomado como protocolo de aplicación. A partir de esto, se comienza a definir la arquitectura orientada a recursos, teniendo en cuenta el proceso de aprendizaje.

RESUMEN EJECUTIVO

La problemática a resolver en este trabajo de grado está dada por dos aspectos fundamentales, el primero de ellos se basa en que los sistemas de e-learning actuales están diseñados con un proceso de enseñanza aprendizaje rígido que debe ser seguido por el usuario, y que no es controlado por el mismo sino por el sistema, de tal manera que brinda poca flexibilidad para el autoaprendizaje; por otra parte las arquitecturas de estos sistemas no soportan características como hipermedia basada en vínculos, ya que por lo general son sistemas orientados a servicios en donde lo que hacen es invocar operaciones que contienen datos pero no información suficiente para dar flexibilidad al autoaprendizaje.

Para poder dar solución a esta problemática, se comenzó realizando una investigación acerca de la creación de un curso en contextos e-learning, de tal manera que fue posible determinar cuáles son los elementos principales que se involucran en este proceso. Posteriormente se definió un proceso de enseñanza/aprendizaje adecuado para el contexto, y se definió como protocolo de aplicación. Una vez culminada esta parte, se comenzó a definir la arquitectura propuesta con base al proceso de enseñanza/aprendizaje, siguiendo una serie de pasos ya establecidos. Culminada la arquitectura, se procede a realizar la validación correspondiente mediante la implementación de un prototipo funcional tomando como caso de estudio una de las asignaturas del Departamento de Ingeniería de Sistemas.

Entre las conclusiones resultantes del desarrollo de este trabajo de grado, se encuentran la disminución de código fuente gracias al uso de la hipermedia entre los recursos, por otra parte el traspaso de links entre recursos facilita la navegación entre los mismos, teniendo en cuenta el protocolo de aplicación definido para el contexto. Otra conclusión que surge en este trabajo de grado se basa en las diferentes representaciones en las que puede presentarse un recurso para el cliente.

INTRODUCCIÓN

La hipermedia es el paso de vínculos entre representaciones de recursos que permiten cambiar el estado de ellos y así mismo de la aplicación, por medio de un proceso de negocio al que en éste contexto se le conoce como protocolo de aplicación, que permite alcanzar un objetivo mediante una serie de pasos ya estructurados. Un protocolo de aplicación consiste de ciertos servicios que tienen asociados a ellos recursos que se encuentran conectados entre sí según un conjunto de interacciones legales ya definidas; de igual modo dichos recursos pueden estar conectados con otros recursos pertenecientes a otros servicios, todo depende de cómo haya sido definido el protocolo de aplicación.

Por otra parte, los avances tecnológicos han permitido que el aprendizaje vaya mucho más allá que la presencia en aulas académicas, razón por la cual una modalidad formativa ha estado tomando fuerza durante los últimos años, se trata del e-learning el cual proporciona una manera diferente de aprender utilizando elementos tecnológicos, entre los cuales se encuentran aplicaciones basadas en procesos de aprendizaje permitiendo que el espacio tiempo ya no sea un inconveniente. Por este motivo se puede decir que no sólo se hablaría de aprendizaje independiente sino también de una ayuda didáctica que puede ser utilizada en aulas académicas.

Teniendo en cuenta lo mencionado anteriormente, se pretende realizar un diseño arquitectural orientado a recursos haciendo uso de la hipermedia (como principio básico en las arquitecturas orientadas a recursos y los servicios web tipo REST), definiendo un proceso de aprendizaje adecuado para e-learning y utilizado como protocolo de aplicación. Una vez generado este diseño, se realizará la respectiva validación mediante un prototipo que maneje contenidos de aprendizaje representados como OVAS (objetos virtuales de aprendizaje), los cuales serán tratados como recursos y serán vinculados entre ellos teniendo en cuenta el protocolo de aplicación que dirige a la arquitectura propuesta.

I - DESCRIPCION GENERAL DEL TRABAJO DE GRADO

1. Oportunidad, Problemática, Antecedentes

1.1 Descripción del contexto

En contextos académicos, se requiere que el acceso a contenidos de aprendizaje sea virtual y para esto, dichos contenidos deben ser creados y almacenados en repositorios como primera medida. Posteriormente se debe encontrar la forma de representarlos correctamente, los cuales se convierten en entidades de negocio que son representadas como recursos.

Con los diferentes avances tecnológicos en la web, los procesos de aprendizaje han cambiado y hoy en día se da la posibilidad de implementar dichos procesos a través de e-learning, es decir diseñar, poner en práctica y evaluar un curso o plan informativo desarrollado, por medio de redes de computadores (Andrea M, 2009), es por esto que los sistemas de gestión de aprendizaje tienen la posibilidad de ofrecer este tipo de aprendizaje. Para que esto sea posible es necesario tener en cuenta el proceso que será utilizado para evaluar a cada consumidor del servicio, como por ejemplo un proceso de aprendizaje. Dicho proceso será tomado como un protocolo de aplicación, es decir como un conjunto de interacciones legales necesarias para llegar a un objetivo (Jim Webber, 2010), en otras palabras se podría decir que será tomado como un proceso de negocio.

Construir aplicaciones basadas en e-learning es algo que se utiliza hoy en día en cualquier institución educativa, ya que es un medio que apoya el aprendizaje por parte del estudiante y que de igual manera, puede ser de apoyo para el docente. Estas aplicaciones, manejan diferentes contenidos académicos, los cuales son en su mayoría didácticos y son una forma diferente de fomentar el aprendizaje (Andrea M, 2009). Pero para construir estas aplicaciones, se debe tener en cuenta los requerimientos que debe satisfacer para que de esta manera se logre identificar la alternativa que mejor se adapte a lo que se desea.

Por otra parte, muchas de estas aplicaciones existentes son hipermediales, es decir, manejan diferentes contenidos como imágenes, sonido, videos, entre otros y su conexión entre ellos (hipervínculos), pero no tienen un proceso de aprendizaje definido, es por esto que cuando se trata de aplicaciones hipermediales, la alternativa que se debe usar es REST ya que se acopla

mejor por el manejo que proporciona a aquellos recursos que en este trabajo de grado serán mencionados como objetos virtuales de aprendizaje, y de igual manera permite establecer un proceso de aprendizaje, el cual en éste Trabajo de Grado será tomado como protocolo de aplicación, es aquí donde entra a jugar un papel importante las arquitecturas orientadas a recursos puesto que uno de los principios de estos dos últimos términos mencionados es la hipermedia (Savas Parastatidis, 2010), que por definición establece que es la herramienta que permite comunicar e interactuar a los usuarios mediante el uso de hipervínculos en diferentes tipos de contenido (Jim Webber, 2010). Teniendo en cuenta lo ya mencionado, se puede decir que aplicar esto es una alternativa nueva para diseñar e implementar aplicaciones hipermediales que estén basadas en un estilo arquitectural y que expongan protocolos de aplicación en contextos e-learning e integren la hipermedia (Savas Parastatidis, 2010).

Ahora es importante tener en cuenta que actualmente existen varios tipos de sistemas de gestión de aprendizaje que manejan contenidos de aprendizaje virtuales de diferentes maneras y están diseñados con diferentes estilos arquitecturales según sea la necesidad de cada uno, es aquí en donde entra una comparación entre las arquitecturas tipo SOA y REST. Una arquitectura tipo SOA (Arquitectura Orientada a Servicios) es una forma lógica de diseño de un sistema de software que permite proporcionar servicios a cualquiera de las aplicaciones de usuario final o para otros servicios distribuidos en la red (Richard N. Taylor, 2010). Mientras que las arquitecturas tipo REST (Representational State Transfer), hacen referencia a una colección de servicios para el diseño de arquitecturas en red y son ideales para sistemas hipermedia distribuidos como lo es la web (Marsset, 2006).

Con estas breves definiciones, se debe aclarar cuando es útil hacer diseños arquitecturales tipo SOA o REST. Si lo que se desea es establecer una descripción formal de cómo es la interfaz del servicio, abordar requerimientos no funcionales y poco complejos (transacciones, seguridad, direccionamiento, entre otros) manteniendo información contextual y estados conversacionales, y centrado en el diseño de aplicaciones distribuidas, la solución que mejor se acomoda es un estilo arquitectural tipo SOA; pero si por lo contrario, lo que se quiere es un servicio web que no mantenga estado, mejor rendimiento y conocimiento del contenido que va a ser comunicado entre consumidor y proveedor, la solución arquitectural que soporta esto debe ser tipo REST (Marsset, 2006). Aun así es necesario aclarar las características correspon-

dientes a cada uno, las arquitecturas tipo SOA generan ciertos requisitos para el intercambio de mensajes como es la implementación de SOAP (Simple Object Access Protocol) que por definición es un protocolo de mensajería estándar usado en servicios web, el cual por lo general utiliza HTTP como medio de transporte para comunicar mensajes entre cliente y servidor.

Aterrizando un poco los estilos arquitecturales tipo REST, existen las arquitecturas orientadas a recursos (ROA), que se basan en representaciones de recursos y acceso a éstos por medio de sus correspondientes URI (Universal Resource Identifier), ésta arquitectura usa el protocolo HTTP no como medio de transporte como en el caso de SOAP, sino como medio de interacción entre el cliente y el servidor, es decir que gracias a este protocolo se puede tener acceso a los recursos por medio de los métodos de éste protocolo; ROA maneja características específicas como lo es el direccionamiento que por definición dice que “una aplicación es direccionable si expone sus datos o los aspectos interesantes de estos como recursos teniendo en cuenta la URI (Universal Resource Identifier) de cada uno de ellos” (ruby, 2007); escalabilidad se enfoca en que las peticiones realizadas al servidor sean únicas, es decir cada vez que el cliente realiza una petición ésta termina donde empezó (ruby, 2007); interoperabilidad se refiere a las formas de representación según el dispositivo para que el recurso pueda ser consumido, también depende del estado del recurso y es mostrado por el servidor en xml, página web o texto separado por comas, puesto de HTTP es un protocolo que no maneja estados, cuando se utiliza de manera adecuada permite interpretar cada mensaje sin tener conocimiento alguno de los mensajes precedentes (Marset, 2006); interface única ya que ROA sólo usa el protocolo HTTP, usa los 4 métodos básicos para el manejo de recursos (GET, PUT, POST, DELETE) (ruby, 2007).

Con esto, se puede decir que los objetos virtuales de aprendizaje pueden ser tratados como recursos en un sistema de gestión de aprendizaje (requieren tener tus propias URI's para poder ser representados y trabajar con dicha representación) (Marset, 2006), posteriormente se pueden crear aplicaciones hipermediales teniendo en cuenta un protocolo de aplicación y así mismo tener en cuenta la novedad de contextos e-learning que permiten que los procesos de aprendizaje sean diferentes. Se puede decir que es una alternativa diferente para crear aplicaciones basadas en recursos (Jim Webber, 2010).

1.2 Formulación del problema que se resolvió

¿Cómo construir sistemas distribuidos hipermediales que expongan protocolos de aplicación para contextos e-learning basados en una arquitectura orientada a recursos?

1.3 Justificación

Manejar objetos virtuales de aprendizaje como recursos en una arquitectura orientada a recursos, permite que el acceso a las representaciones de éstos sea más ágil gracias al uso del protocolo HTTP y de sus métodos. Pero el acceso a estos recursos debe estar definido en un orden según reglas establecidas, por esto la idea de un protocolo de aplicación basado en procesos de aprendizaje permite que esto sea posible, ya que al establecer un conjunto de interacciones entre los recursos (Jim Webber, 2010) hace que sea posible modificar el estado de la aplicación. Para que esto sea posible, es necesario escoger un diseño arquitectural que se acomode a lo que se quiere lograr, teniendo en cuenta lo mencionado anteriormente el estilo más adecuado es una arquitectura orientada a recursos, puesto que sus ventajas permiten una facilidad en el diseño (interoperabilidad, rendimiento e interfaz única).

Cabe aclarar que SOA permite también desempeñar esta labor mencionada anteriormente, pero es un estilo arquitectural que se acomoda de mejor manera en otro tipo de contextos y aplicaciones, y de igual manera el intercambio de objetos virtuales de aprendizaje por medio de SOAP implica empaquetar el mensaje en un documento XML teniendo en cuenta la estructura de los sobres (envelope), encabezados (header) y cuerpo del mensaje (body) (Papaoglou, 2008), y enviar todo esto a través de HTTP. Ya que este Trabajo de Grado se centra en la idea del uso de una arquitectura orientada a recursos para construir aplicaciones distribuidas, es importante aclarar las ventajas de este estilo arquitecturas, que en otras palabras vienen siendo las misma que REST, por esto se aclara que dicho estilo se enfoca en escalabilidad y rendimiento para este tipo de sistemas y tiene como ventaja que maneja muchos recursos por medio de pocas operaciones ya definidas por el protocolo HTTP, mientras que SOA se centra en el diseño de dichas aplicaciones y maneja muchas operaciones pero con escasez de recursos (Marset, 2006). Es importante aclarar que crear una aplicación que exponga protocolos de aplicación y que además, se base en procesos de aprendizaje permite al cliente extender y facilitar el acceso a la información no sólo a una persona sino de manera

colectiva de tal forma que sea de gran ayuda para aquellos que no pueden estar de manera presencial adquiriendo conocimientos en alguna asignatura o tema, dicha persona(s) que utilicen el servicio incrementará la autonomía y responsabilidad teniendo en cuenta el proceso de aprendizaje establecido y permite que las limitaciones de espacio/tiempo se hagan a un lado (Andrea M, 2009). De igual manera, permite que el aprendizaje logrado por el estudiante en una clase presencial, sea complementado por medio de aplicaciones hipermediales.

1.4 Impacto Esperado

El diseño arquitectural que se genere en este proyecto tiene como fin a corto, mediano y largo plazo, ser una base para la creación de plataformas que se encuentren en contextos e-learning de manera que aprovechen el uso de la hipermedia junto con objetos virtuales de aprendizaje. A corto plazo, se pretende que se use en el desarrollo de aplicaciones que se basen en procesos de aprendizaje en contextos de e-learning, teniendo en cuenta la creación de objetos virtuales de aprendizaje y el correcto uso de estos en una arquitectura orientada a recursos.

A mediano plazo y a largo plazo, se pretende que la arquitectura propuesta sea parte de una plataforma de aprendizaje que haga parte del departamento de ingeniería de sistemas, de manera que se convierta en un apoyo para estudiantes en las aulas de clase y por fuera de estas.

Se tiene como meta principal, que el diseño arquitectural sea un apoyo para desarrolladores de aplicaciones y/o plataformas en contextos e-learning, manejando un protocolo de aplicación establecido junto con objetos virtuales de aprendizaje de modo que se generen diferentes alternativas, que apoyen el aprendizaje hacia los estudiantes.

2. Descripción del Proyecto

2.1 Visión global

El desarrollo de aplicaciones para e-learning se ha convertido en un aspecto fundamental en aulas de clase, la implementación de estas basadas en un diseño adecuado permite conocer a fondo cómo este tipo de aplicaciones dan soporte al aprendizaje. La arquitectura orientada a recursos que se propuso en este trabajo de grado se basa en un proceso de aprendizaje propuesto para el aprendiz, quien a su vez lo dirige según sus necesidades. Este trabajo de grado se desarrolló teniendo en cuenta un proceso de aprendizaje, que incluye recursos conectados

entre sí permitiendo una navegabilidad basada en la hipermedia, de manera que se aprovechó una de las características de las arquitecturas tipo REST y el estilo arquitectural orientado a recursos.

2.3 Objetivo general

Construir una arquitectura orientada a recursos que soporte un protocolo de aplicación en el dominio de e-learning.

2.4 Fases Metodológicas o conjunto de objetivos específicos

Las fases metodológicas que se manejaron para el desarrollo de este proyecto se describen a continuación.

Fase 1: Caracterización

Esta fase fue alineada de acuerdo al siguiente objetivo específico: caracterizar el estilo arquitectural orientado a recursos en contextos de aprendizaje.

Principalmente se realizaron actividades de identificación de elementos relevantes que hacen parte de temas correspondientes a una arquitectura orientada a recursos enfocados en procesos de aprendizaje en e-learning y objetos virtuales de aprendizaje.

Fase 2: Diseño

Esta fase fue alineada de acuerdo a los siguientes objetivos específicos: diseñar un protocolo basado en recursos hipermediales para una aplicación e-learning; diseñar una arquitectura dirigida por recursos que soporte el protocolo diseñado.

Para el desarrollo de esta fase se realizaron actividades de levantamiento de requerimientos arquitecturalmente significantes, diseño de protocolo de aplicación y de una arquitectura orientada a recursos teniendo en cuenta el contexto de e-learning y el protocolo diseñado.

Fase 3: Validación

Esta fase fue alineada de acuerdo al siguiente objetivo específico: Validar la arquitectura mediante la implementación de un prototipo en el dominio de e-learning. Durante esta fase se realizaron actividades relacionadas con el levantamiento de requerimientos propios del prototipo, desarrollo y publicación del mismo con su respectiva documentación.

2.5 Método que se propuso para satisfacer cada fase metodológica

Ya que para la realización de éste trabajo de grado, se implementó una metodología SCRUM, cada fase metodológica fue tomada como un sprint, en donde fue necesario aplicar un conjunto de métodos que permitieran alcanzar cada uno de los objetivos y así cumplir el objetivo general de éste trabajo de grado.

1. Caracterización

Para esta fase se propuso utilizar la metodología de investigación especulativa o deductiva, la cual permite que a través de conceptos se genere una estructuración teórica, que da paso a adaptar los elementos identificados en el contexto de una arquitectura orientada a recurso, teniendo en cuenta e-learning como el área en la que se trabajó.

2. Diseño

Esta fase se desarrolló utilizando una metodología de investigación inductiva, la cual toma como base los resultados obtenidos en la fase anterior. Con esto se dio paso a diseñar un protocolo de aplicación en base a un proceso de aprendizaje adecuado para contextos e-learning, a partir de esto se diseñó una arquitectura orientada a recursos que soportara tanto objetos virtuales de aprendizaje como el protocolo de aplicación diseñado.

3. Validación

Para la elaboración del prototipo se implementó una metodología SCRUM, en donde por medio de iteraciones cortas fue posible realizar la validación de las fases anteriores creando una aplicación basada en los resultados que se obtuvieron en las fases anteriores. Se establecieron 2 sprints principales (construcción de objetos virtuales de aprendizaje y construcción de la aplicación) que tendrán en cuenta los resultados finales de las fases anteriores para poder desarrollar las etapas correspondientes a ésta última fase.

II - MARCO TEÓRICO

1. Marco Contextual

Para el trabajo de grado que se propone, se abarcarán conceptos como hipermedia, e-learning, protocolo de aplicación y arquitectura orientada a recursos, los cuales relacionados de distintas maneras permiten conocer estudios realizados en el área que de una u otra manera, aportan conocimiento del manejo que se le ha dado a aplicaciones que se encuentren basadas en los conceptos ya mencionados. A continuación se presentan algunos de los estudios que se han realizado en el área de arquitectura orientada a recursos incorporando conceptos como los ya mencionados.

Uno de los estudios mencionados se presenta en el artículo titulado: “*Resource Oriented Architecture for Business Processes*” (Xiwei, Liming, Yang, & Staples, 2010), el cual muestra la relación entre una arquitectura orientada a recursos y un proceso de negocio, indicando tres pasos esenciales para representar el proceso en la arquitectura mencionada: 1. Mapear los conceptos de proceso, caso, estado, evaluación y enrutamiento, en recursos; 2. Idear un conjunto de mensajes tipo request/responde con sus respectivos formatos para cada tipo de recurso; 3. Designar la semántica estándar de métodos HTTP para invocar a los recursos. Este estudio concluye indicando que los servicios web tipo RESTful colocan los principios de la web en sistemas con procesos intensivos, además propone un modelado de proceso que relaciona diferentes conceptos utilizando los principios de REST y de una arquitectura orientada a recursos (ROA).

Otro artículo titulado: “*Rest Based Web Access to Learning Design Services*” (Dodero & Ghiglione, 2008), aclara en qué consiste el acceso a servicios de aprendizaje basados en el estilo REST, explicando a su vez los beneficios que esto proporciona tales como el acceso a través de una interfaz uniforme, accesibilidad a los recursos, interconexión entre recursos y extensibilidad. Este estudio concluye afirmando que una solución simple a través de REST que se encuentre bien definida y basada en los diferentes principios de éste estilo proporciona operaciones bien definidas, escalables, basadas en protocolos tipo stateless y estados distri-

buidos, y separado por capas permite cualquier número de intermediarios entre las actividades y los servicios.

En el estudio titulado: “*Developing web services choreography standards – the case os REST vs. SOAP*” (Michael zur Muehlana, 2005), presenta un caso de estudio en el desarrollo de estándares en el área de flujos de trabajo entre organizaciones basadas en servicios web. Ya que estos facilitan la comunicación entre sistemas manteniendo la promesa de integración automatizada y la coordinación a través de las fronteras institucionales, los principios de REST han trabajado en estos estándares asociados con una nueva versión de SOAP (usado sin violar los principios de REST). Además resalta a los participantes que se encuentran involucrados en la decisión de REST o SOAP tal como lo son los usuarios (quienes consumen el servicio web) los cuales pueden encontrarse dentro de la compañía que desarrolla el servicio web, y los vendedores quienes optan una posición por algún estándar según el contexto en el que se encuentre. Este estudio concluye afirmando que la decisión de una compañía en usar servicios web tipo REST o SOAP, se delega a nivel técnico en donde los argumentos que se presenten generarán un debate que proporcionará argumentos suficientes para tomar esta decisión.

1.1 REST en sus comienzos

Se requería de un modelo que permitiera ver como WWW (World Wide Web) debería trabajar, así que se idealizó un modelo que permitiera visualizar las interacciones en una aplicación web, el cual dio paso a la creación de una arquitectura web moderna que proveía una serie de principios los cuales permitirían identificar fallas existentes en una arquitectura (ROY T. FIELDING, 2002). A partir de esto, se definió REST como “un conjunto coordinado de restricciones arquitectónicas que intenta minimizar la latencia y la comunicación de red, mientras que al mismo tiempo maximiza la independencia y la escalabilidad de las implementaciones de componentes” (ROY T. FIELDING, 2002). La primera versión de REST surgió entre octubre de 1994 y agosto de 1995 como respuesta a la comunicación entre conceptos web mientras se desarrollaba la versión 1.0 de HTTP y se daba el primer paso de la versión 1.1. La idea original se refería a un “modelo de objetos HTTP” pero este nombre

hacía mayor énfasis en un servidor HTTP, así que con el fin de proporcionar más información acerca del comportamiento de una aplicación web bien diseñada, se optó por otorgar el nombre de “Representational State Transfer” en donde se habla acerca de una máquina de estados generada por una red de páginas web, permitiendo así al usuario avanzar a través de una aplicación seleccionando un enlace o enviando un formulario de entrada de datos definido, cada acción representa una transición de estado de la aplicación por medio del paso de una representación de dicho estado para el usuario (ROY T. FIELDING, 2002). Es por esto que hoy en día la web se considera una instancia del estilo arquitectural REST, aunque algunas aplicaciones web basadas en este estilo arquitectural pueden incluir acceso a otros tipos de interacción, pero en énfasis principal de REST y su desarrollo es la hipermedia distribuida, por lo cual sólo se centra en partes arquitecturales que son consideradas esenciales para la interacción distribuida hipermedial.

1.2 REST vs SOAP

La siguiente tabla muestra las diferencias principales entre REST (Representational State Transfer) y SOAP (Simple Object Access Protocol) (Pautasso, 2007).

Diferencia	REST	SOAP
Principal	<ul style="list-style-type: none"> • “La web es el universo de información accesible a nivel mundial”. (Tim Berners LEE). • En REST las solicitudes deberán publicar sus datos en la web (a través de la URIs). La siguiente imagen ilustra el diseño de un servicio tipo REST, en donde se aprecia el uso de métodos HTTP y acceso a los recursos por medio de URIs. 	<p>A diferencia de REST, SOAP utiliza la web como transporte universal de mensajes, por tal motivo transporta dichos mensajes por medio de métodos HTTP. La siguiente imagen ilustra el diseño de un servicio tipo SOAP, en donde se puede apreciar el protocolo por el cual es transmitido el mensaje, el método HTTP utilizado, y el destino al cual se dirige</p>

	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">POX (Plain Old XML)</div> <div style="display: flex; justify-content: space-around; margin-bottom: 5px;"> <div style="border: 1px solid black; padding: 2px; text-align: center;">HTTP GET</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">HTTP POST</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">HTTP PUT</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">HTTP DEL</div> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;"><i>Resource</i> URI</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">Application</div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">SOAP (WS-*)</div> <div style="display: flex; justify-content: space-around; margin-bottom: 5px;"> <div style="border: 1px solid black; padding: 2px; text-align: center;">SMTP</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">HTTP POST</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">MQ...</div> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;"><i>Endpoint</i> URI</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">Application</div>
<p>Conceptual</p>	<p>REST utiliza la web como medio de publicación abierta, además está orientado a recursos razón por la cual maneja acceso informacional a estos. Está diseñado para recibir cualquier cantidad de clientes.</p>	<p>SOAP utiliza la web como medio de comunicación cruzado entre empresas, administra los procesos de negocio para dirigirlos al consumidor, interactúa directamente con los recursos. Por otra parte los servicios que son implementados en SOAP son para el comercio entre empresas generalmente, por lo cual maneja un conjunto seleccionado de clientes teniendo en cuenta visibilidad y acceso controlado por contratos.</p>
<p>Tecnológica</p>	<p>Utiliza pocas operaciones ya que maneja una interfaz única, por tal motivo tiene la capaci-</p>	<p>Maneja un flujo de eventos confiables, pero al mismo tiempo utiliza muchas operaciones ya que no ma-</p>

	<p>dad de manejar muchos recursos, identificándolos por medio de URIs. REST está centrado en escalabilidad y rendimiento de los sistemas distribuidos a gran escala hipermedia. La interacción está dirigida por el usuario por medio de formularios.</p>	<p>neja una interfaz única sino que depende de la interfaz del servicio, por este motivo no tiene la capacidad de manejar muchos recursos, los cuales no manejan un mecanismo de nomenclatura estándar. SOAP está centrado en el diseño de aplicaciones integradas (distribuidas) y está compuesto por procesos de negocio</p>
Servicio	<p>Se basa en información legible para las personas que definen URIs de petición y respuesta. Al momento de interactuar con el servicio, implica grandes cantidades de tiempo en pruebas y depuración de URIs manualmente construidas como combinación de parámetros. Maneja WADL como descripción del servicio, pero este mecanismo fue propuesto en 2008 y aún no ha sido estandarizado</p>	<p>Los “stubs” en los clientes pueden ser construidos a partir del WSDL en la mayoría de lenguajes de programación. Se requiere que cada servicio publique su propia interfaz con diferente semántica. La versión 1.1 del WSDL en donde todo tipo de puerto puede ser enlazado a métodos HTTP u otros protocolos. La versión más reciente del WSDL es la 2.0, la cual es más flexible ya que cada operación puede seleccionar el método que utiliza.</p>
Diseño	<p>Se requiere de los siguientes pasos para diseñar un servicio REST:</p>	<p>Se requiere de los siguientes pasos para diseñar un servicio SOAP:</p>

	<ul style="list-style-type: none"> • Identificar los recursos a ser expuestos como servicios. • Definir “buenas” URLs para direccionarlos. • Distinguir recursos de solo lectura y libres de efectos colaterales (GET) de recursos modificables (POST, PUT, DELETE). • Relaciones entre recursos correspondientes a hipervínculos que puedan ser seguidos para obtener más detalles. • Implementar y desplegar en el servidor web. 	<ul style="list-style-type: none"> • Listar qué son operaciones de servicio en el WSDL del servicio. • Definir un modelo de datos para el contenido de los mensajes intercambiados a través del servicio. • Seleccionar un protocolo de transporte adecuado para enlazar los mensajes de la operación y definir la calidad del servicio, seguridad y políticas transaccionales. • Implementar y desplegar en el contenedor del servicio web.
Estado	<p>Provee transiciones de estado explícitas, el servidor es stateless, los recursos contienen datos y vínculos representando transiciones válidas de estado. Los clientes mantienen correctamente el estado siguiendo los</p>	<p>Los servicios SOAP tienen transiciones de estado implícitas, los servidores podrían mantener conversaciones de estado a través de múltiples intercambios de mensajes, los mensaje sólo</p>

	<p>enlaces de forma genérica. Utiliza tres técnicas para agregar sesiones al protocolo HTTP: Cookies (HTTP Headers), re escribir URL, campos de formulario ocultos.</p>	<p>contienen datos pero no incluyen información hacer de transiciones de estado válidas, los clientes mantienen el estado adivinando la “máquina de estado del servicios”. Utiliza una técnica para agregar sesiones al protocolo SOAP: cabeceras de sesión.</p>
--	---	--

Tabla 1. REST vs SOAP

2. Marco Conceptual

Los temas principales que se van a tratar en esta sección son hipermedia, REST, ROA (Arquitectura Orientada a Recursos), y e-learning. Los conceptos de REST son fundamentales para para entender los conceptos de ROA, estos presentan un principio fundamental común que es la hipermedia sobre la cual se fundamenta el HATEOAS que es concepto sobre el cual gira el presente trabajo de grado. Estos conceptos se aterrizan en el contexto de e-learning.

Para ejemplificar los conceptos se tomó como base el caso de estudio de presentado en la siguiente referencia: (Jim Webber, 2010).

2.1 Hipermedia

Conjunto de métodos o procedimientos para escribir, diseñar o componer contenidos que integren soportes tales como: texto, imagen, video, audio, mapas y otros soportes de información emergentes, de tal modo que el resultado obtenido, tenga la posibilidad de interactuar con los usuarios (Jim Webber, 2010).

2.1.1 HATEOAS (Hypermedia as the engine of application state)

Un sistema hipermedia se caracteriza por la transferencia de links entre el intercambio de representaciones de recursos por los participantes en un protocolo de aplicación. Dichos enlaces anuncian otros recursos que se encuentran en el protocolo de aplicación. Los enlaces a menudo mejoran con el marcado semántico para dar significados de dominio a los recursos que se identifican (Jim Webber, 2010). Este intercambio de enlaces entre representaciones de recursos, genera un cambio en el estado de la aplicación, lo cual se considera como el resultado del comportamiento de todo: el servicio, el cliente, el intercambio de las representaciones de recursos hipermedia disponibles, y la publicidad y selección de enlaces.

En cada interacción, el servicio y el consumidor intercambian representaciones del estado de un recurso, no el estado de la aplicación. Una transferencia de representación incluye enlaces que reflejan el estado de la aplicación. Estos enlaces legítimos anuncian las transiciones de estado de aplicación, pero el estado de la aplicación no se encuentra explícito en la representación recibida por el consumidor, este es inferido por el consumidor basado en el estado de todos los recursos (potencialmente distribuido a través de muchos servicios) con el cual el consumidor está interactuando actualmente, en otras palabras el estado de la aplicación está dado por el cambio de estado de los recursos.

El estado actual de un recurso es una combinación de:

- Los valores de elementos de información que pertenecen a ese recurso.
- Enlaces relacionados a los recursos.
- Enlaces que representan una transición a un posible estado futuro del recurso actual.
- El resultado de evaluar cualquier regla de negocio que se relacione con el recurso o con otros recursos locales.

El último punto anterior hace referencia a que el estado de un recurso depende en parte del estado de los otros recursos. De igual forma las reglas que controlan el estado del recurso son internas al servicio que lo rige, no están disponibles para el consumidor, es decir, el estado del recurso es una función de un conjunto privado de reglas que solo son conocidas por el recurso, estas reglas no se pierden en una representación externa (Jim Webber, 2010).

Las reglas de negocio que relacionan a un recurso con otros recursos sólo deben referirse a recursos propios locales. Siempre es posible organizar los recursos propios locales para evitar dependencias locales, pero no se puede realizar lo mismo si los recursos asociados hacen parte de otro servicio.

Un servicio hace cumplir un protocolo de aplicación anunciando las interacciones legítimas con los recursos pertinentes. Cuando un consumidor sigue enlaces integrados en representaciones de recursos y subsecuentemente interacciones con recursos vinculados, todo el estado de la aplicación cambia. Los vínculos entre recursos permiten que al seguir un protocolo de aplicación se modifique el estado de un recurso y al mismo tiempo el de la aplicación.

Los consumidores de sistemas hipermedia causan transiciones de estado visitando y manipulando el estado del recurso. El estado de la aplicación cambia como resultado de un consumidor que es dirigido por un sistema hipermedia, el cual es similar a un proceso de negocio. Esto sugiere que un servicio puede anunciar flujos de trabajo usando hipermedia (Jim Webber, 2010).

La hipermedia hace sencillo implementar protocolos de negocio (también nombrados en este documento como protocolos de aplicación), de manera que reduce el acoplamiento entre servicios y consumidores. Sin necesidad de entender la estructura de una URI (Uniform Resource Identifier), el consumidor solo requiere entender la estructura o contexto de negocio en donde aparece el vínculo. Esto reduce la dependencia de la aplicación con datos semánticos como templates de URIs o una descripción del lenguaje web de la aplicación (WADL) (Jim Webber, 2010).

2.1.2 Formatos Hipermedia

Los sistemas hipermediales colocan demandas similares a los consumidores tal como lo hace la web con las personas: los consumidores necesitan descubrir e interactuar con los recursos para poder realizar el objetivo de la aplicación. Para ilustrar como la representación de formatos permite a los consumidores descubrir e interactuar con los recursos, se pondrá como ejemplo XHTML, el cual es uno de los formatos más representativos y populares en la web. XHTML es usado para representar información en una página y para vincular otras páginas o

contenidos (es la descripción del protocolo). La inclusión de vínculos a otros recursos hace de XHTML un formato hipermedia (Jim Webber, 2010). La web es creyente a los formatos de representación intercambiados por los consumidores y los servicios, lo cual es una de las principales razones de su éxito en diversos dominos. Pero cuando se trata de hipermedia, no todos los formatos son iguales.

2.1.3 Hipermedia Dead Ends

A pesar del éxito y de las ventajas de los formatos hipermedia en la web, las aplicaciones hoy en día no los usan y simplemente se limitan a formatos planos como XML para la integración de sistemas. Aunque XML es fácil de usar como formato de intercambio de datos, es absolutamente obvio en la web. No hay absolutamente nada malo con la representación XML cuando es considerada por separado, ya que transmite correctamente el estado actual de la orden. Pero este no proporciona ningún contexto, lo cual significa que no indica ningún estado actual en un proceso de negocio, o cómo este avanza (Jim Webber, 2010).

El uso de XML deja al consumidor sin una guía (protocolo) para cumplir de manera satisfactoria la transacción de negocio que ha sido indicada, ya que no hay un control hipermedia en la representación, de esta forma el consumidor debe confiar en la información que se encuentra fuera para determinar que hacer a continuación. Desde el punto de vista de bajo acoplamiento, esta es una decisión de un diseño mal elaborada, ya que no contiene una estructura sólida y completa. Los aspectos de la implementación del servicio se filtran a través de mecanismos tales como templates de URIs en la ejecución de los consumidores, haciendo el cambio difícil y arriesgado (Jim Webber, 2010).

También es posible, comunicar el protocolo de información con desarrolladores de la aplicación consumidor usando documentación escrita o contratos estáticos como WSDL (Web Service Description Language), WADL (Web Application Description Language), o templates de URIs.

2.1.4 Templates de URIs y acoplamiento

Primero es necesario considerar como se debería comunicar el protocolo de información si se escoge de manera estática. Es posible compartir el template de URI con los consumidores. Por ejemplo, se puede documentar y compartir el template `http://restbucks.com/payment/{order_id}`, la documentación describe como los consumidores esperarían colocar una representación de una URI generada reemplazando `{order_id}` por el ID original de la orden. Con una pequeña manipulación de la URI el trabajo puede hacerse rápidamente, pero así mismo puede fallar (Jim Webber, 2010).

Generalmente, es mejor exponer URIs estables, estas actúan como puntos de servicio para que después la hipermedia se haga cargo. Continuando con el ejemplo de órdenes de servicio, un punto de entrada estaría dado por `http://restbucks.com/order` de tal manera que se interactúa con el recurso en esa URI que genera otras gestiones de recursos, cada una genera vínculos hipermedia a más recursos que se encuentren vinculados en el proceso de negocio de órdenes (Jim Webber, 2010).

2.1.5 Selección de un formato hipermedia

Los formatos proporcionan los medios para interactuar con un servicio y son parte del contrato de servicio, por esta razón es importante escoger el formato hipermedia apropiado al momento de diseñar. En hipermedia no se da una representación específica de un formato, pero se requiere un formato capaz de transmitir información hipermedia necesaria. Diferentes formatos hipermedia se adaptan a diferentes servicios, escoger depende del trade-off entre alcance y utilidad.

2.1.5.1 Formatos hipermedia estándar

Varios de los formatos hipermedia que se encuentran en uso en la web, hoy en día tienen la capacidad de soportar algunos de los requerimientos más comunes. Formatos como ATOM (y RSS) y XHTML son extensamente utilizados y entendidos. Correspondientemente, muchas herramientas de software y librerías están disponibles para producir, consumir, y administrar representaciones de recursos en estos formatos (Jim Webber, 2010).

2.1.5.2 Formatos hipermedia de dominio específico

Ya que la web es escéptica a las representaciones de formatos, es posible crear formatos personalizados que se adapten a problemas de dominio específicos. Si se usa un formato altamente entendido o se crea uno propio, los formatos hipermedia son más amigables a la web que aquellos que no lo son (Jim Webber, 2010). Un buen formato hipermedia combina dominio específico y protocolo de información. La información de un dominio específico incluye valores de elementos de información que pertenecen a un recurso de negocio. El protocolo de información declara como hacer seguimiento al progreso en un proceso de negocio.

En un formato hipermedia, los controles hipermedia representan un protocolo de información lo cual incluye dirección del vínculo del recurso, junto con algunas marcas semánticas. En el contexto actual de la representación del recurso, las marcas semánticas indican el significado del vínculo del recurso.

Crear un formato hipermedia de dominio específico no es tal difícil como parece ser, se construye en base de un esquema XML que ya ha sido creado, lo siguiente es introducir representaciones de los elementos adicionales. Es posible añadir controles hipermedia a estos esquemas definiendo ambos vínculos y la marca semántica necesaria.

Una forma de agregar controles hipermedia es distinguir entre la acción de vincular y la acción de añadir significado a los vínculos. Vincular es proceso repetitivo. Los significados atribuidos a los enlaces cambian de un contexto a otro. Para esto se define un elemento `<link>` para transmitir el dominio a la función del vínculo, y un atributo `rel` para representar la semántica de la aplicación asociada a un vínculo. Modificando el ejemplo anterior con lo que acaba de ser mencionado, se presenta la siguiente imagen.

```
<order xmlns="http://schemas.restbucks.com">
  <location>takeAway</location>
  <item>
    <name>latte</name>
    <quantity>1</quantity>
    <milk>whole</milk>
    <size>small</size>
  </item>
  <cost>2.0</cost>
  <status>payment-expected</status>
  <link rel="http://relations.restbucks.com/payment"
    href="https://restbucks.com/payment/1234" />
</order>
```

Ilustración 1 Formato Hipermedia vinculando y añadiendo vínculos

Añadiendo el elemento `<link>` al esquema, se puede decir que se ha definido con éxito un formato hipermedia propio.

2.1.6 Procesando formatos hipermedia

Una vez definidos los formatos hipermedia, es necesario transmitir a los consumidores como procesar y la razón acerca de ésta representación. Afortunadamente HTTP provee una manera de identificar una representación particular como los tipos de media, utilizando como cabecera el content-type.

2.1.6.1 Tipos de media

Un tipo de media especifica una interpretación del esquema de las representaciones de recursos, este esquema describe en términos de codificación, esquemas, y modelos de proceso, lo cual es una clave para la creación de un contrato de servicio (Jim Webber, 2010).

El nombre de un tipo de media (como aplicación/xml) es clave en la interpretación del esquema del tipo de media. Los tipos de media son declarados en las cabeceras HTTP con content-type, ya que se encuentran por fuera de la carga útil del body, los cuales están relacionados, los consumidores pueden determinar cómo procesar una representación sin tener que primero revisar la carga útil e inspeccionar su contenido.

Un valor del tipo de media indica el esquema preferido por el servicio al momento de interpretar una representación: los consumidores son libres de ignorar esta recomendación y procesar la representación como mejor les parezca.

Los tipos de media son uno de los 3 componentes claves de un protocolo de aplicación de dominio, los otros dos son vínculos relacionados con valores, los cuales describen los roles de los recursos vinculados, e idiomas HTTP, los cuales administran a los recursos que participan en el protocolo. Lo hacen indicando el rol al recurso vinculado en el contexto actual de la representación. Un valor del tipo de media ayuda al consumidor a entender qué es el final de un vínculo, cómo la interacción con el recurso es realizada con métodos HTTP, y las alternativas de control sugeridas por los códigos del estatus HTTP (Jim Webber, 2010).

2.1.7 Protocolos Hipermedia

REST introduce un conjunto de principios que al ser aplicados en el diseño de una aplicación, produce características deseables como escalabilidad, uniformidad, desempeño y encapsulamiento. Usar HTTP, URIs, e hipermedia se pueden construir sistemas que exhiban exactamente las mismas características. Estos tres términos permiten implementar protocolos de aplicación adaptados a soluciones de necesidades de negocio (Jim Webber, 2010).

Un protocolo de dominio de aplicación resume las peticiones HTTP soportadas por el servicio asociadas a un flujo de trabajo lógico que cada petición activa.

Cuando se pasa del diseño a la implementación, es necesario pensar en el protocolo de una manera diferente. En una aplicación distribuida orientada a recursos, el protocolo puede ser pensado como una función de uno o más ciclos de vida de un recurso y reglas de negocio que conectan a estos recursos (Jim Webber, 2010). Cualquier flujo de trabajo en la implementación del servicio está relacionado con el ciclo de vida de los recursos, no del ciclo de vida del protocolo de aplicación.

2.2 REST (Representational State Transfer)

Los servicios web son sistemas software diseñados para soportar una interacción interoperable máquina a máquina sobre una red, esta definición alberga muchos tipos diferentes de

sistemas, pero el caso común de uso de refiere a clientes y servidores que se comunican mediante mensajes XML que siguen el estándar SOAP (Marset, 2006).

En los últimos años se ha popularizado un estilo de arquitectura de Software conocido como REST (Representational State Transfer). Los Servicios Web basados en REST intentan imitar al protocolo HTTP o protocolos similares mediante la restricción de establecer la interfaz a un conjunto conocido de operaciones estándar (por ejemplo GET, PUT,...). Por tanto, este estilo se centra más en interactuar con recursos con estado, que con mensajes y operaciones (Marset, 2006). REST (Representational State Transfer) es un estilo de arquitectura de software para sistemas hipermedias distribuidos tales como la Web. El término frecuentemente es utilizado en el sentido de describir a cualquier interfaz que transmite datos específicos de un dominio sobre HTTP sin una capa adicional, como hace SOAP. Es importante aclarar que REST no es un estándar, ya que es tan solo un estilo de arquitectura. Aunque REST no es un estándar, está basado en estándares: HTTP, URL, Representación de los recursos: XML/HTML/GIF/JPEG/, Tipos MIME: text/xml, text/html.

2.2.1 Recurso

Un recurso es algo que puede ser almacenado en un computador y representado como un conjunto de bits: un documento, una fila en una base de datos, o el resultado de la ejecución de un algoritmo (Marset, 2006). Un recurso puede ser un objeto físico como una manzana, o un concepto abstracto como el valor.

2.2.1.1 URIs

Para que un recurso pueda ser considerado como tal debe tener al menos una URI, la cual es el nombre y la dirección del recurso. Una URI nunca debe representar a más de un recurso. Si un pedazo de información no tiene una URI, entonces no es un recurso y no se encuentra en la web, a excepción de un trozo de datos que describa a otro recurso. Por otra parte las URIs deben ser descriptivas, por lo cual deben tener una estructura (Marset, 2006). Algunos Ejemplos:
<http://www.lesoir.be>; <http://www.lesoir.be/edition/20-01-2012>;
<http://www.example.org/newspapers/belgium>

La siguiente imagen refleja la estructura de una URI, en donde se ven los componentes servicio, método y argumentos:

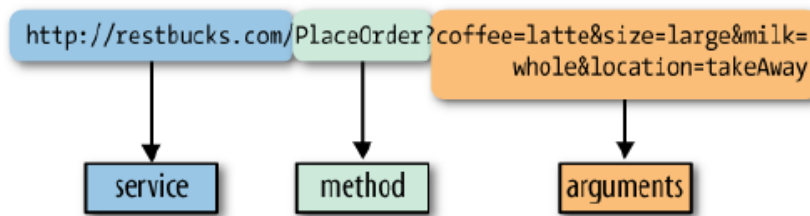


Ilustración 2 Estructura de una URI

2.2.1.2 Representaciones

Es la descripción de (o una parte de) un recurso, este puede tener múltiples representaciones text, XHTML, XML, JSON. En REST, tanto cliente como servidor intercambian representaciones de recursos por medio de una interfaz y protocolo definido (Fielding, 2000).

Para cada recurso es necesario especificar: Las representaciones que el servicio ofrece al cliente y las representaciones que el servicio acepta del cliente. En lo posible se deben utilizar representaciones estándares.

2.2.1.3 Negociación de contenido

Es el proceso de selección de la mejor representación para una respuesta dada cuando se tienen múltiples representaciones disponibles. Puede ser usada para obtener distintas representaciones del mismo recurso (ruby, 2007).

Ejemplos: <http://www.example.org/newspapers/belgium> y
<http://www.example.org/newspapers/belgium.json>

2.3 Arquitectura Orientada a Recursos (Resource Oriented Architecture)

Conjunto de guías de diseño y de buenas prácticas que pueden ser utilizadas para diseñar servicios web tipo RESTful. Las siguientes son las propiedades de ROA:

2.3.1 Adressability

El servicio web debería hacer la aplicación lo más localizable posible, es decir que cada pieza de información de la aplicación debería ser un recurso y tener una URI, haciendo que el recurso sea fácil de buscar. Esta es la única pieza de información que se necesita publicar para hacer que un recurso sea accesible. Las URIs hacen a los recursos enlazables ya que estos están expuestos a través de una interfaz uniforme (ruby, 2007).

2.3.2 Interfaz Uniforme

El protocolo web de facto es HTTP, es un documento basado en un protocolo request/response estandarizado entre cliente y servidor. HTTP es la interfaz uniforme de servicios web tipo RESTful (ruby, 2007).

Los servicios web construidos en SOAP, WSDL, y otros WS* también utilizan HTTP como medio de transporte, pero utilizan solo unas pocas de sus capacidades. Los servicios web tipo RESTful tienen una interfaz uniforme (métodos HTTP y URIs), así que una vez conocida la ubicación del recurso (por medio de su URI), es posible invocar el método HTTP (GET, POST, ect.).

Una interfaz única provee interoperabilidad entre las aplicaciones. HTTP es extensamente soportado, y el número de librerías cliente HTTP garantiza que no habrá inconvenientes de comunicación. Todos los accesos a los recursos ocurren a través de la interfaz uniforme de HTTP (GET, POST, PUT, DELETE, HEAD, OPTIONS). A continuación se ilustran los significados de los métodos HTTP:

CRUD	REST	
CREATE	POST	Create a (sub)resource
RETRIEVE	GET	Retrieve a representation of a resource
UPDATE	PUT	Modify a resource/create a new resource
DELETE	DELETE	delete a resource
	OPTIONS	Discover what HTTP methods are supported by the resource
	HEAD	requests headers only (similar to GET but omits representation)

Ilustración 3 Métodos HTTP y sus significados

2.3.3 Connectedness

Los promotores de REST afirman que los recursos deben estar conectados tanto como sea posible (ruby, 2007). A continuación se muestra un ejemplo de la representación de un CD conectado con otros recursos por medio de la etiqueta <link>.

```
<cd>
  <title>Ella and Louis</title>
  <year ref="http://music.com/year/1956">1956</year>
  <artist ref="http://music.com/artists/123">Ella Fitzgerald</artist>
  <artist ref="http://music.com/artists/456">Louis Armstrong</artist>
  <link rel="self" type="text/json" href="http://music.com/album/789"/>
  <link rel="self" type="text/xml" href="http://music.com/album/789"/>
  <link rel="http://music.com/album/comments" type="text/xml" ↩
    href="http://music.com/album/789/comments"/>
</cd>
```

Ilustración 4 Representación de un CD conectado con otros recursos

En la ilustración 4, el cliente puede cambiar el estado de la aplicación por medio de comentarios en el álbum. La lista de comentarios del álbum es un recurso accesible por medio de la URI <http://music.com/album/789/comments> ya que este servicio web utiliza una interfaz única, el cliente conoce el formato de la URI, los tipos de contenido disponibles y el formato de datos, sabrá exactamente como interactuar con él. Con el método GET recupera la lista de comentarios existentes, y con el método POST crea un nuevo comentario. A partir de una única petición inicial, el cliente puede tomar muchas decisiones: la hipermedia impulsa el estado de la aplicación (Jim Webber, 2010).

Otro aspecto importante de la hipermedia es el estado de la aplicación, el cual es llevado por esta. El hecho que el servicio web provea un conjunto de links, permite al cliente cambiar el estado de la aplicación simplemente a través del siguiente link. En la siguiente imagen se ilustra el cambio de estado de la aplicación mediante el paso de un recurso a otro por medio de sus URIs.

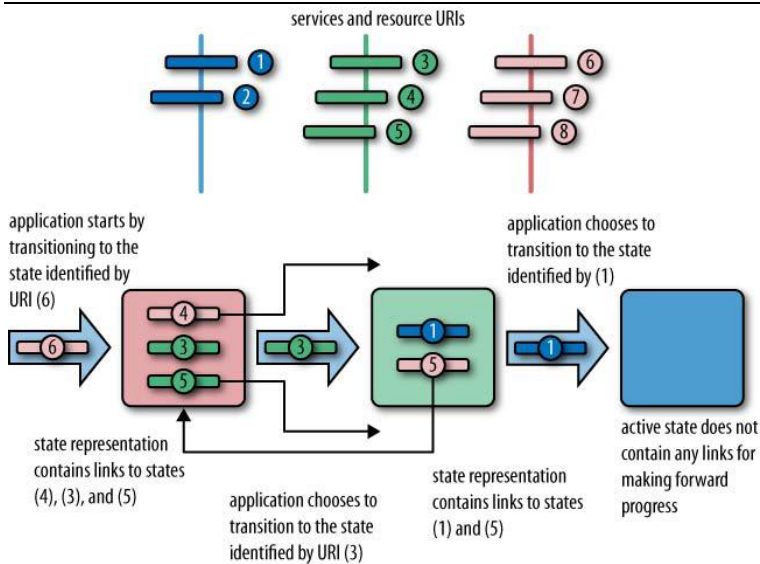


Ilustración 5 Cambio de estado de la aplicación

2.3.4 Statelessness

Cada petición del cliente al servidor debe contener toda la información necesaria para entender la petición, y no puede tomar ventaja de cualquier estado del cliente almacenado en el servidor. El estado de la aplicación debe mantenerse en el cliente, pero el estado del recurso debe estar almacenado en el servidor (Jim Webber, 2010). Mejora la confiabilidad ya que facilita la recuperación de fallos parciales, de igual forma mejora la escalabilidad ya que el servidor puede liberar rápidamente recursos de computación, y diferentes peticiones pueden ser atendidas por diferentes servidores (balanceo de cargas) ya que el servidor no administra el uso de los recursos a través de las peticiones. El statelessness viene con una ventaja como la escalabilidad ya que no maneja ninguna información de sesión, no se necesita enrutar los subsiguientes requerimientos al mismo servidor, manejo de fracasos, entre otros.

2.3.5 Idempotency

La idempotencia es una propiedad que al aplicar la misma operación varias veces, produce siempre el mismo resultado (ruby, 2007). GET, HEAD, OPTIONS son operaciones de sólo lectura pero PUT, POST, DELETE son operaciones de lectura/escritura con efectos colaterales. La idempotencia y las operaciones de sólo lectura pueden ser ejecutadas múltiples veces

de manera segura sin riesgo de errores (ruby, 2007). La operación POST no es idempotente ni de sólo lectura, por lo cual no es seguro re ejecutarla.

2.4 E-learning

Es una modalidad formativa que es realizada a distancia y que utiliza como soporte nuevas tecnologías. Este concepto se encuentra incluido en la teleformación, en donde ésta es considerada una de las alternativas para la formación a distancia, puesto que surge como una solución para aquellos individuos que como primera medida por su situación geográfica, condiciones de trabajo y/o físicas, no pueden acceder a la formación con carácter presencial.

Por otra parte, el e-learning supone 5 ítems fundamentales:

- Separación alumno-profesor.
- Utilización de semántica de medios y recursos técnicos.
- Aprendizaje individual.
- Apoyo de una organización de carácter tutorial.
- Comunicación bidireccional.

Los cuales conllevan a un listado de características fundamentales, tales como:

- Utilización de diferentes herramientas de comunicación tanto sincrónica como asincrónica.
- Almacenaje, mantenimiento y administración de los materiales sobre un servidor web.
- Uso de protocolos tcp y http para facilitar la comunicación entre los estudiantes y los materiales de aprendizaje o los recursos.
- Autoaprendizaje mediado por ordenador.
- Uso de navegadores web para acceder a la información.
- Aprendizaje flexible.
- Materiales digitales.

De igual manera surgen otro tipo de características que son tratadas como ventajas y que se encuentran fuertemente relacionados con el contexto de e-learning:

- Autoaprendizaje debido a las necesidades individuales y sociales del sujeto.

-
- Flexibilidad en el espacio y en el tiempo. Desaparecen las barreras espacio-temporales.
 - Cuenta con múltiples fuentes informativos en cualquier momento (enlaces web).
 - Posibilidad de evaluación continúa.
 - Interactividad. Los alumnos pueden comunicarse entre sí, con el formador y con los recursos online disponibles en internet.

Ya que el e-learning en pocas palabras es una forma diferente de aprendizaje, existen modelos pedagógicos para que el proceso de enseñanza-aprendizaje sea adecuado para esta nueva modalidad. Ya que e-learning se fundamenta en el uso de tecnologías, los modelos pedagógicos en este contexto se encuentran basados en procesos formativos en teleformación. A continuación se explican los 3 modelos que mejor se acomodan:

- Tecnologías transmisivas: se caracterizan por presentar un carácter más lineal debido a la interacción alumno-profesor. Carece de ambiente de aprendizaje.
- Tecnologías interactivas: se basa en definir el sistema por el cual se aprende a acceder a la información que se pretende transmitir. Intervienen computadores, programas de enseñanza asistida por este y los productos multimedia.
- Tecnologías colaborativas: se centra en la introducción de espacios colaborativos.

NOTA: En el siguiente capítulo se profundiza en los conceptos de e-learning, ya que estos son fundamentales para el desarrollo de este trabajo de grado.

III – DESARROLLO DEL TRABAJO

En esta sección del documento se desarrolla el tema central del trabajo de grado: la arquitectura Orientada a Recursos para un curso de e-learning. La siguiente figura ilustra el proceso metodológico seguido para la construcción de dicha arquitectura orientada a recursos.



Ilustración 6 Proceso de construcción de una arquitectura orientada a recursos en e-learning

El primer paso es el proceso de construcción de un curso e-learning en donde se define su estructura en base a los elementos que hacen parte de este (Objetivos de aprendizaje y actividades correspondientes). Posteriormente se procede a definir el protocolo de aplicación correspondiente en base a un proceso de enseñanza/aprendizaje, relacionando los elementos del curso construido. Finalmente con los resultados de los pasos anteriores, se realiza el diseño de una arquitectura orientada a recursos que soporte el diseño de protocolo de aplicación.

3.1 Proceso de construcción de un curso de e-learning

Esta sección mostrará la construcción de un curso de e-learning, que posteriormente permitirá soportar la definición del protocolo de aplicación.

Para la construcción de un curso de e-learning se necesita un modelo de diseño instruccional que pueda ser usado para definir las actividades de desarrollo de un curso. El diseño instruccional es el desarrollo semántico de especificaciones usando teorías de aprendizaje e instruccionales para asegurar la calidad del aprendizaje (Nations, 2011).

Existen muchos sistemas de diseño de modelo instruccional, la mayoría están basados en el modelo popular ADDIE, que consta de cinco fases: análisis, diseño, desarrollo, implementación y evaluación.

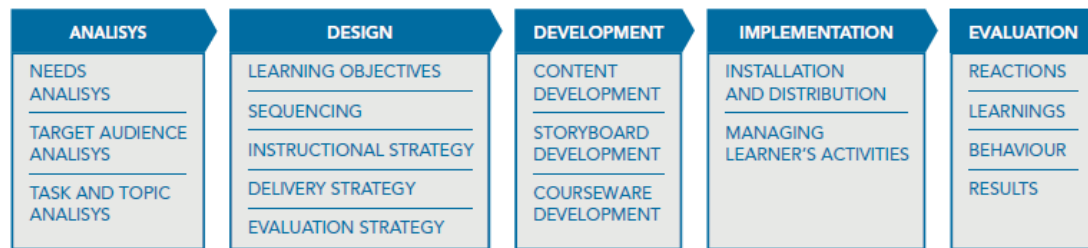


Ilustración 7 Modelo ADDIE

3.1.1 Análisis

El análisis de necesidades permite la identificación de objetivos del curso generales de alto nivel. El análisis de la audiencia influencia el diseño y despliegue del e-learning. El análisis también es necesario para determinar el contenido del curso. Se utilizará el análisis de tópicos para identificar y clasificar el contenido del curso (Nations, 2011).

3.1.1.1 Identificando el contenido del curso

De acuerdo al Modelo ADDIE (Fase de análisis – Análisis de Tópicos) en esta actividad se identifica el contenido detallado del curso para alcanzar el objetivo general. El análisis del contenido es el paso más crítico en el proceso de diseño instruccional, dicho análisis es un pre requisito para desarrollar objetivos de aprendizaje específicos y el esquema del plan de estudios (Nations, 2011).

El análisis e identificación de contenido se realizará aplicando el siguiente método:

- Análisis de tópicos: permite identificar y clasificar el contenido del curso. Es apropiado para cursos que son diseñados principalmente para proveer información o lograr alcanzar objetivos educativos.
- Identificar el contenido del curso: se pretende identificar las principales categorías de contenido para el curso. En este trabajo de grado se tomará como base los syllabus (temarios), los cuales proveen las categorías de contenido y sus detalles.
- Clasificar elementos de contenido: permite reconocer las conexiones a través de los elementos de contenido para refinar el esquema del curso. Los elementos de contenido pueden ser clasificados de acuerdo a los tipos de contenido que ellos representan.

Varias clasificaciones han sido desarrolladas por varios autores de acuerdo a necesidades específicas. La clasificación presentada aquí es una versión simplificada de Morrison, Kemp & Ross (2001) (Nations, 2011), que identifica seis tipos principales de contenido como se muestra en la siguiente tabla:

Tipos de elementos de contenido	
Hechos	Únicos, información específica que responde a las preguntas: quién, dónde, cuándo. Los hechos son mostrados, exhibidos o indicados. Ejemplos: datos, listas, historial de eventos.
Procedimientos	Un procedimiento es una serie de pasos claramente definidos, con el objetivo de realizar una tarea. Un procedimiento responde a la pregunta: cómo. Ejemplo: “instrucciones para crear una tabla en Microsoft Word”.
Conceptos	Un concepto es un grupo de objetivos, entidades o ideas que: son definidas por una palabra o término único; comparten características comunes; difieren en características poco importantes; requieren una definición; y responden a la pregunta: qué es. Ejemplo: El concepto de “cambio de clima”.
Principios	Un principio (o regla) describe una relación entre dos conceptos. Por ejemplo: “como el precio aumenta, aumenta la oferta”. Algunos principios pueden ser traducidos en guías estratégicas las cuales pueden guiar decisiones y tareas complejas. Ejemplo: “Directrices para enfrentar la volatilidad de los precios”.
Herramientas interpersonales	Herramientas verbales o no verbales para interactuar con otras personas. Ejemplo: contenido relacionado con “negociación” o “resolver un conflicto de grupo”.
Posiciones	Comportamiento de predisposición. Ejemplo: contenido relacionado con apreciar “la importancia y urgencia de adoptar medidas para limitar los impactos negativos del cambio

	de clima”.
--	------------

Tabla 1 Tipos de Contenidos, Morrison, Kemp & Ross

3.1.2 Diseño

De acuerdo al Modelo ADDIE La fase de diseño abarca las siguientes actividades:

- Formulación de un conjunto de objetivos de aprendizaje que se requieren para alcanzar los objetivos de alto nivel.
- Definir el orden en el cual los objetivos deberían ser alcanzados, esto se verá reflejado en el protocolo de aplicación.
- Seleccionar estrategias instruccionales de evaluación y de despliegue del curso.

La salida de la fase de diseño es un modelo que podría ser usado como referencia para desarrollar el curso. El modelo ilustra la estructura del curso (unidades, lecciones, actividades), los objetivos de aprendizaje asociados con cada unidad y los métodos de despliegue y formatos para desplegar cada unidad (Nations, 2011).

3.1.2.1 Definición de objetivos de aprendizaje

Un objetivo de aprendizaje es una “sentencia que describe una competencia o capacidad de rendimiento para ser adquirido por el aprendiz” (Nations, 2011). Los objetivos deberían ser especificados para el curso como también para cada actividad.

Basados en lo elementos de contenido identificados en el análisis de tópicos, es posible transformar el objetivo del curso, en objetivos más específicos de aprendizaje.

Los objetivos de aprendizaje combinan dos elementos principales:

- Nivel esperado de desempeño (a través de un verbo de acción como “describir” o “explicar”).

- Contenido de aprendizaje (el tipo de conocimiento o herramientas que deben ser aprendidas).

De acuerdo a la taxonomía de Bloom para el dominio cognitivo (Nations, 2011), los objetivos de aprendizaje pueden implicar seis diferentes tipos de rendimiento cognitivo como se muestra en la siguiente tabla:

Niveles de rendimiento para el dominio cognitivo	
Recordar	El aprendiz es capaz de reconocer o memorizar información
Entender	El aprendiz es capaz de reformular un concepto
Aplicar	El aprendiz es capaz de usar información en nuevas formas
Analizar	El aprendiz es capaz de descomponer y definir relaciones entre componentes
Evaluar	El aprendiz es capaz de justificar una decisión de acuerdo a unos criterios o estándares
Crear	El aprendiz es capaz de crear un nuevo producto o alternativa

Tabla 2 Niveles de rendimiento para el dominio cognitivo

Los objetivos de aprendizaje también podrían estar relacionados con otras taxonomías incluyen los dominios afectivos y psicomotores, sin embargo en este trabajo de grado se limitarán al dominio cognitivo (Nations, 2011).

3.1.2.2 Definición de las actividades

Una actividad se encuentra alineada con un objetivo de aprendizaje, y para cumplirlo es necesaria la ejecución de un conjunto de actividades.

Ya que existen varias formas de utilizar el e-learning como herramienta de aprendizaje, se han identificado formas discretas de actividades de aprendizaje que apoyan este tipo de aprendizaje (Ron Oliver, 2000). Los entornos que se encuentran basados en e-learning se pueden describir como parte de un proceso de mapeo utilizando las prácticas de enseñanza tradicionales tal como lo muestra la siguiente tabla.

Componentes	Descripción
General	Inscribirse en un curso, leer el plan de estudios, lectura del material del curso
Lecturas	Lecturas y conferencias presenciales
Discusiones en grupo	Participar en discusiones de grupo y sesiones tipo seminario
Eventos de aprendizaje	Salidas de campo, actividades prácticas, lecturas adicionales
Comunicación	Comunicación privada entre instructor y compañeros
Auto estudio	Práctica supervisada, lecturas no supervisadas, pequeñas tareas
Proyectos individuales	Principales tareas del curso
Proyectos grupales	Tarea del curso completada colaborativamente
Exámenes	Actividades de tarea

Tabla 3 Componentes de enseñanza tradicional

Ya que los componentes de enseñanza tradicional presentados en la tabla 3, se acomodan para el e-learning, se presenta un modelo para describir este tipo de aprendizaje identificando formas funcionales discretas que pueden ser asumidas por entornos de e-learning (Ron Oliver, 2000). Las cuatro categorías que se han identificado son acceso a la información, comunicaciones en red, aprendizaje interactivo y desarrollo de materiales. Cada una de éstas categorías describe un atributo único y tiende a utilizar aplicaciones de la web para proporcionar oportunidades al e-learning (Ron Oliver, 2000). La siguiente tabla describe estas categorías y provee un ejemplo de las tecnologías soportadas por cada una.

Estrategia de aprendizaje	Actividad de aprendizaje	Ejemplos
Acceso a la información	La web se utiliza para transmitir información por sí sola para el alumno, por ejemplo el programa de una asignatura, calendario de clase, descripción de tareas, lecturas, descripción de talleres, etc.	Archivos PDF, documentos HTML, audios, videos, archivos de aplicaciones.

Aprendizaje interactivo	La web se utiliza para involucrar elementos de instrucción que se dedican al aprendiz, fomentar la reflexión y toma de decisiones, y proporcionar información en respuesta a las acciones de los educandos.	Java Applets, animaciones, películas.
Comunicaciones en red	La web se utiliza para proporcionar un medio para que la organización, la comunicación y el intercambio de ideas e información entre alumnos y los profesores y otras partes del proceso de aprendizaje.	Email, chat, discusiones de grupo
Desarrollo de materiales	Cuando la web se utiliza como un medio para que los aprendices creen y publiquen materiales, esta se utiliza como una herramienta para la recolección y recopilación de información y la presentación de esta en un formulario publicado.	Crear y publicar páginas web, contenido web y procesos FTP.

Tabla 4 Estrategias de aprendizaje

Para este trabajo de grado, la definición de las actividades para cada objetivo de aprendizaje estará basada en las tablas 3 y 4.

Las actividades que se definirán en este trabajo de grado estarán compuestas por objetos virtuales de aprendizaje como recurso de apoyo para el aprendiz, que estarán relacionados con las estrategias de aprendizaje representadas en la tabla 4.

3.1.2.3 Verificando la alineación de objetivos de aprendizaje y actividades

Objetivos de aprendizaje claros permiten el desarrollo de actividades de aprendizaje que están enfocadas en las necesidades de los aprendices.

Las actividades deben estar alineadas con los objetivos de aprendizaje, esto es deben permitir ayudar a desarrollar el desempeño y contenido de aprendizaje expresado en los objetivos de aprendizaje.

Para este trabajo de grado, las actividades contarán con un proceso similar al de los objetivos de aprendizaje, lo cual implica que un grupo de actividades estará relacionada con un objetivo de aprendizaje. En las siguientes secciones se irá dando claridad al proceso que tendrán las actividades con respecto a los objetivos de aprendizaje.

3.1.3 Desarrollo

En esta fase, el contenido es realmente producido (Nations, 2011). El contenido puede variar considerablemente dependiendo de la disponibilidad de los recursos. Se puede decir que un contenido de e-learning consiste en materiales simples (aquellos con poca o ninguna interactividad o multimedia, como documentos PDF estructurados), los cuales pueden ser combinados con otros materiales (audio o archivos de video), tareas y evaluaciones.

El desarrollo de contenido interactivo multimedia está compuesto de 3 pasos principales:

- Desarrollo de contenido: escribir o recolectar todo el conocimiento e información requerida.
- Desarrollo del storyboard: integrando métodos instruccionales (todos los elementos pedagógicos necesarios para soportar el proceso de aprendizaje) y los elementos mediales. Esto se hace desarrollando el storyboard, esto es, un documento que describe todos los componentes de los productos interactivos finales, incluyendo imágenes, texto, interacciones, pruebas (Nations, 2011).
- Desarrollo de material de enseñanza informatizado (courseware): se desarrollan los componentes interactivos y mediales, produciendo el curso en diferentes formatos y se integran los elementos de contenido dentro de una plataforma.

En este trabajo de grado no se realizará el paso anterior (Desarrollo de material de enseñanza informatizado).

3.1.4 Implementación

En esta fase el curso es entregado a los aprendices, el courseware es instalado en un servidor haciéndolo accesible para los aprendices. Esta fase no será abordada en este trabajo de grado ya que no hace parte del alcance definido.

3.1.5 Evaluación

Un proyecto de e-learning puede ser evaluado para propósitos específicos de evaluación. Se puede evaluar las reacciones de los aprendices, el logro de objetivos y la transferencia de conocimientos y habilidades. Esta fase no será abordada en este trabajo de grado ya que no hace parte del alcance definido y el tiempo establecido para el desarrollo no permite desarrollar esta fase.

3.1.6 Equipo

La participación en proyectos de e-learning requiere habilidades en ciertas áreas, tales como tecnología e hipermedia, por otra parte los integrantes del equipo podrían tener que cambiar su rol tradicional y ejecutar otras tareas (Nations, 2011). Los siguientes roles son requeridos para ejecutar el modelo de actividades ADDIE.

- Administrador de desarrollo de capacidades de recursos humanos: conduce las etapas de necesidades de necesidades y análisis de la audiencia objetivo.
- Diseñador instruccional: es responsable de la estrategia instruccional, escoge las estrategias instruccionales apropiadas. También es responsable del diseño de las actividades y materiales específicas de e-learning que serán parte del curso, incluyendo el desarrollo del storyboard.
- Experto del tema: contribuyen con el conocimiento e información requerida para un curso.

-
- Desarrolladores web y editores hipermedia: son responsables del desarrollo del curso, desarrollan los componentes interactivos e hipermediales, crean el courseware.
 - Administradores del curso: está involucrado en la fase de implementación.
 - Especialistas técnicos de soporte: soporte técnico a través de todas las fases del proceso.

Estos roles serán asumidos en su totalidad por el estudiante y el director este trabajo de grado.

3.2 Definición del protocolo de aplicación en e-learning

En la siguiente sección se describe el proceso de definición del protocolo de aplicación para un dominio de e-learning, el cual conllevará a identificar los elementos del protocolo de aplicación, esto es componentes de un curso e-learning (Objetivos de aprendizaje y actividades), sus relaciones y su transición de estados. (Corresponde al Modelo ADDIE – Fase de diseño – Secuencia)

Por medio de este protocolo se desea conocer cómo los objetivos de aprendizaje (Ver Sección 3.1.2.1 Definición de objetivos de aprendizaje) deben ser secuenciados cuando se estructura un curso, al igual que las actividades que se encuentran relacionadas para cada objetivo de aprendizaje (Nations, 2011).

La definición del protocolo de aplicación se encuentra enmarcada dentro de la fase de diseño del ADDIE.

El protocolo de aplicación tiene como objetivo definir la secuencia de los objetivos de aprendizaje cuando se estructura un curso, a través de las actividades que soportan dichos objetivos, las cuales también tienen definido un protocolo de aplicación que permite cumplir el objetivo de aprendizaje al que se encuentran relacionadas.

Existen varios métodos para definir la secuencia del curso, uno de estos es el método de pre requisitos (Nations, 2011) el cual utiliza una jerarquía de objetivos de aprendizaje, enseñando primero aquellas herramientas que parecen ser pre requisito de otras. Es posible crear la

jerarquía a través de los objetivos de aprendizaje usando los resultados de análisis de tópicos (Ver sección 3.1.1.1 Identificando el contenido del curso). Este método también aplica para definir la secuencia que deben seguir las actividades, las cuales hacen parte de la jerarquía en la que se encuentran los objetivos de aprendizaje.

Se recomienda utilizar un diagrama como el que se muestra a continuación:

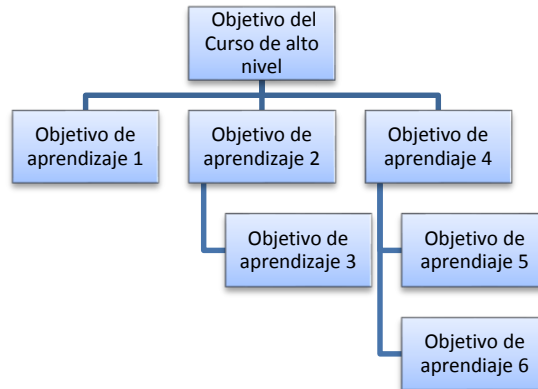


Ilustración 8 Estructura del curso a través de la jerarquía de objetivos de aprendizaje

En la figura anterior el elemento “Objetivo del curso de alto nivel” es igual a como se definió en el análisis de necesidades (Ver Sección 3.1.1 Análisis). Los objetivos de aprendizaje que se encuentran en el mismo nivel son independientes entre ellos, es decir ninguno es pre requisito del otro; en estas condiciones el aprendiz puede decidir tomar cualquiera de las ramas de la jerarquía del curso. Lo anterior implica que el aprendiz es quien podría manejar el estado dentro de su proceso de aprendizaje.

La siguiente figura ilustra de igual forma la estructura del curso a través de la jerarquía de objetivos de aprendizaje, teniendo en cuenta las actividades.

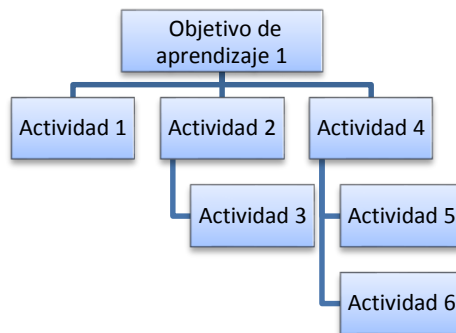


Ilustración 9 Estructura de un curso a través de la jerarquía de actividades

Las actividades que se encuentran en el mismo nivel son independientes entre ellas, es decir ninguna es pre requisito de otra; en estas condiciones el aprendiz puede decidir tomar cualquiera de las ramas de la jerarquía de actividades.

Tomando como ejemplo la ilustración 8, el aprendiz podría tomar los objetivos 1, 2 ó 4 de manera independiente, pero si desea tomar los objetivos 3,5 ó 6, tiene que completar los pre requisitos correspondientes para cada uno de ellos. A partir de esto, se podría generar una máquina de estados en donde se identifiquen los posibles caminos que pueden ser tomados por el aprendiz, los estados corresponderían a los objetivos de aprendizaje y se realizaría una transición de estados cada vez que el aprendiz complete la serie de actividades propuestas para cada objetivo de aprendizaje.

Este mismo ejemplo aplica para la ilustración 9, en donde también se generaría una máquina de estados de los posibles caminos a tomar por parte del aprendiz. Esto implica que el estado de las actividades también sería manejado por el aprendiz durante su proceso de aprendizaje.

La siguiente figura ilustra el diagrama de estados para un objetivo de aprendizaje:

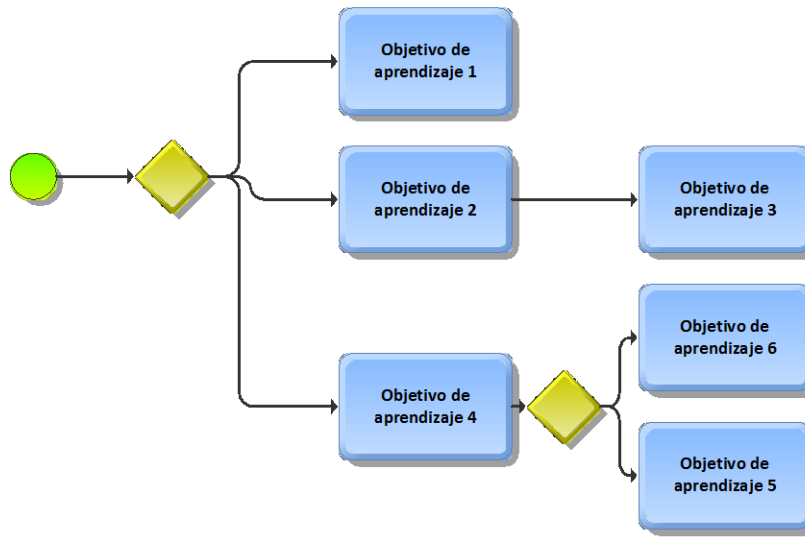


Ilustración 10 Diagrama de estados de objetivos de aprendizaje

Para la ilustración 8, la siguiente imagen representa el diagrama de estados para un solo camino de aprendizaje:

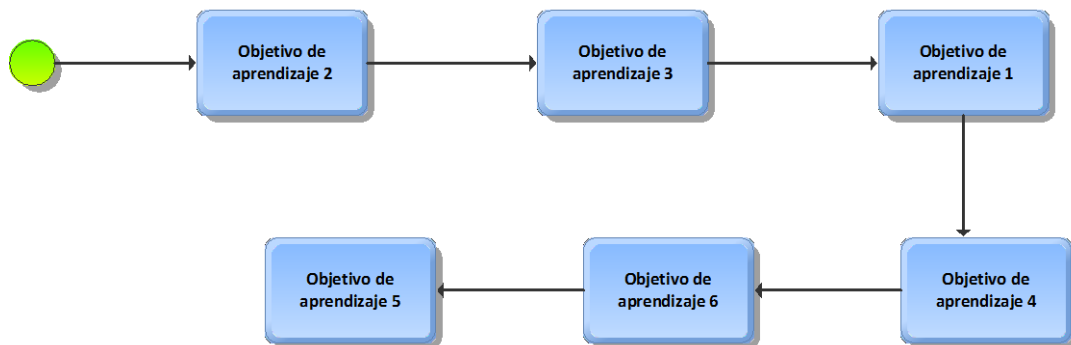


Ilustración 11 Diagrama de estados para un camino de aprendizaje

La siguiente figura ilustra la máquina de estados genérica para las actividades:

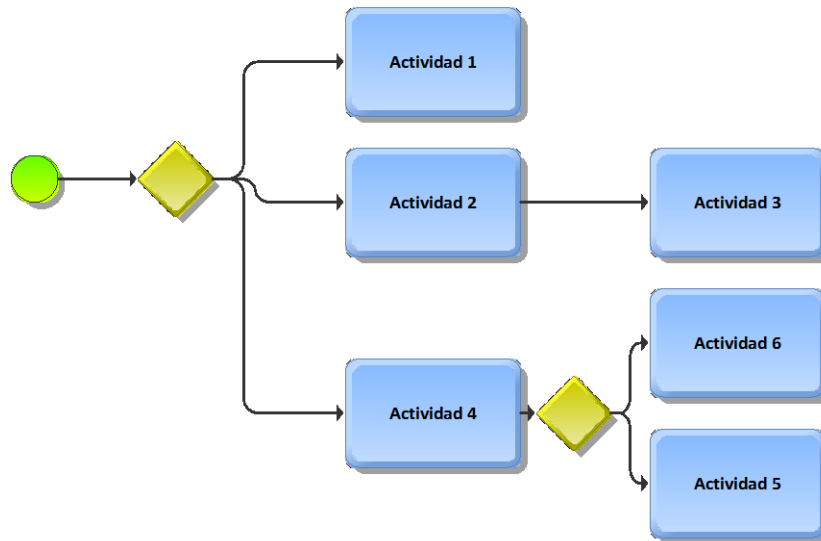


Ilustración 12 Máquina de estados genérica para actividades

La siguiente figura muestra una posible máquina de estados para una actividad, siguiendo un camino de aprendizaje:

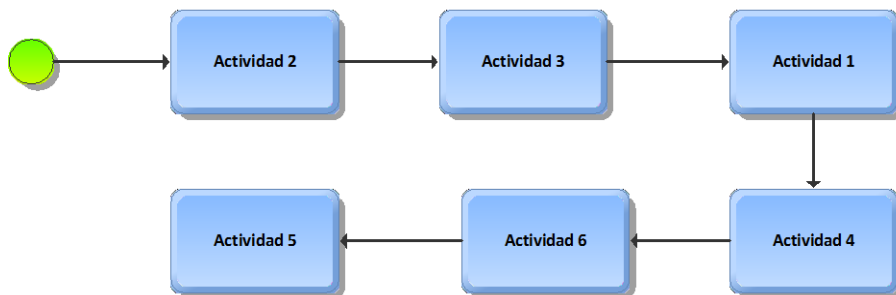


Ilustración 13 Máquina de estados para actividades

La salida del secuenciamiento de las ilustraciones 11 y 13, es la estructura de un curso en la cual cada elemento corresponde a un objetivo de aprendizaje específico y una actividad, lo cual contribuye a alcanzar el objetivo del curso de alto nivel.

Ya que las actividades están compuestas de OVAs (Objetos virtuales de aprendizaje) como se mencionó en el numeral anterior (Ver Sección 3.1.2.2 Definición de las actividades), estos

también manejan una máquina de estados igual a la que se presentó en las ilustraciones 11 y 13. A continuación se presenta la máquina de estados para los objetos virtuales de aprendizaje.

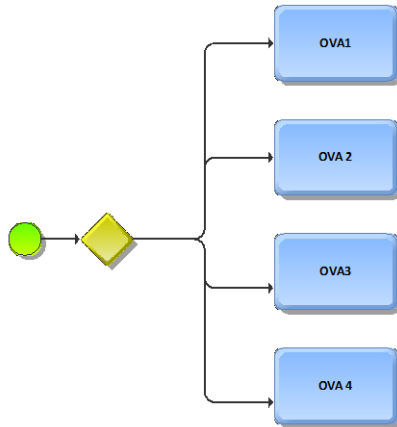


Ilustración 14 Máquina de estados para OVAs

A diferencia de los objetivos de aprendizaje y de las actividades, los objetos virtuales de aprendizaje no son dependientes entre sí, por tal motivo el aprendiz tiene la opción de comenzar a desarrollarlos por donde prefiera hasta terminarlos, completando así la actividad.

La siguiente imagen es un ejemplo de la estructura de un curso de e-learning, el cual puede incluir unidades con diferentes números de sesiones. En un curso de autoaprendizaje, cada sesión es un objetivo de aprendizaje.

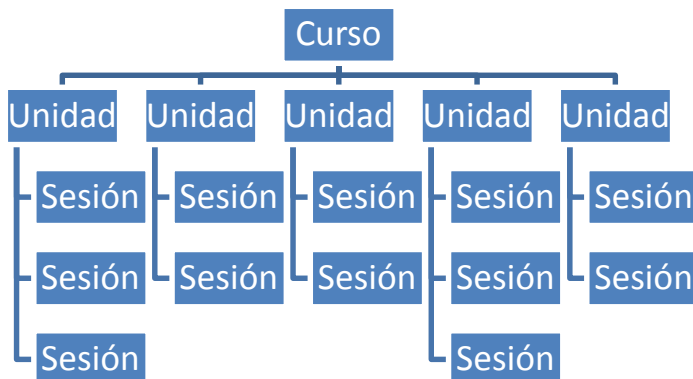


Ilustración 15 Estructura de un curso

En la ilustración anterior, una unidad corresponde a unidades temáticas del curso, la sesión son las unidades de trabajo medidas en tiempo de los aprendices y corresponde a un objetivo de aprendizaje.

El protocolo de aplicación que se propone tiene los siguientes elementos:

- Curso: Asignatura o disciplina académica que se impartirá en el contexto de e-learning.
- Unidad: Como se definió anteriormente.
- Objetivo de aprendizaje: Un objetivo de aprendizaje es una sentencia que describe una competencia o capacidad de rendimiento para ser adquirido por el aprendiz.
- Sesión: Como se definió anteriormente.
- Actividad: Es una tarea que se desarrolla para alcanzar un objetivo de aprendizaje, se desarrolla dentro de una sesión, soportan el proceso de aprendizaje.
- OVA: Son los recursos hipermediales que se presentan a los aprendices para apoyar las actividades.

La siguiente figura representa el modelo de relación entre dichos elementos:

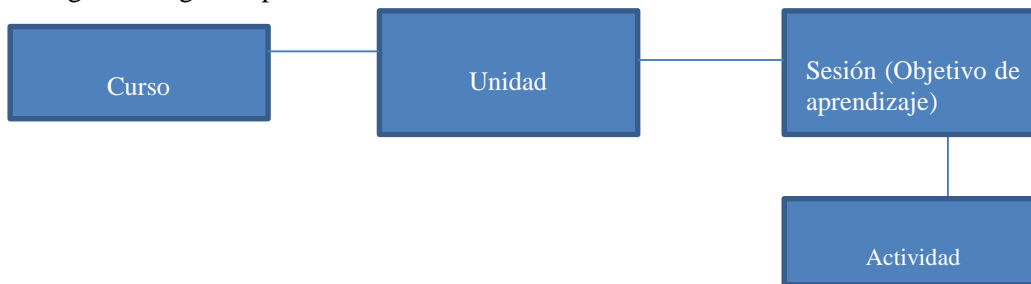


Ilustración 16 Relación curso, unidad, sesión y actividad

De la imagen anterior se aprecian las siguientes relaciones entre los elementos:

- Un curso consta de una unidad.
- Una unidad tiene varias sesiones (objetivos de aprendizaje).
- Una sesión tiene varias actividades.

La siguiente máquina de estado presenta el modelo general del proceso de enseñanza/aprendizaje articulando las máquinas de estado representadas en las figuras 11, 13 y 14, en

donde las relaciones reflexivas indican que cada elemento tiene su propia transición de estados:

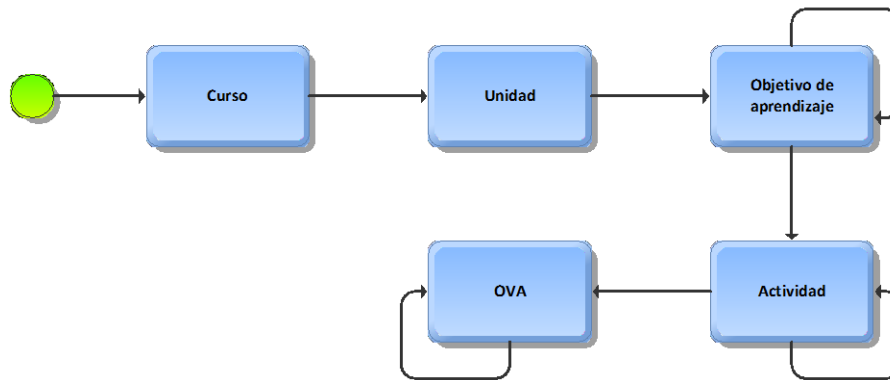


Ilustración 17 Máquina de estado general del proceso

3.2.1 Modelo de transición entre estados

Uno de los elementos fundamentales del protocolo de la aplicación consiste en cómo se realizará la transición entre los estados.

Como se presentó en la sección anterior, existen tres máquinas de estado por cada recurso las cuales manejan su transición de estados como lo ilustra la siguiente figura:



Ilustración 18 Modelo de transición entre estados

En los recursos de tipo OVA para realizar la transición del estado “En Proceso” al estado “Terminado”, se requiere completar un proceso de navegación de cada uno de los elementos constitutivos del OVA (Slides como se verá más adelante) como se muestra en la siguiente figura:

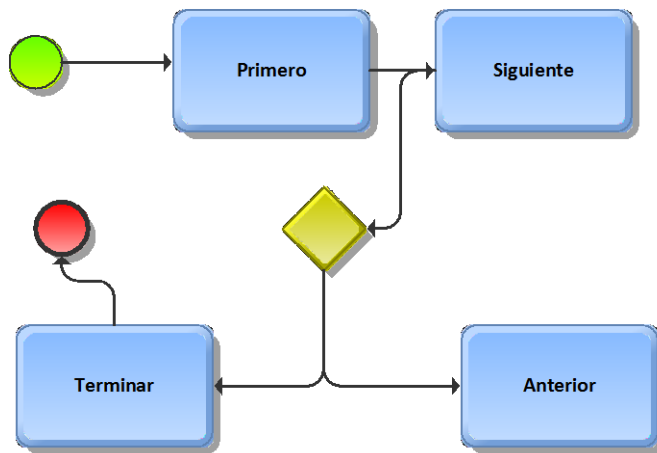


Ilustración 19 Navegación de OVAS

A continuación se presenta la transición de estados entre los recursos:

Inicialmente todos los recursos se encuentran en estado “No Iniciado”, tal como lo representa la siguiente figura

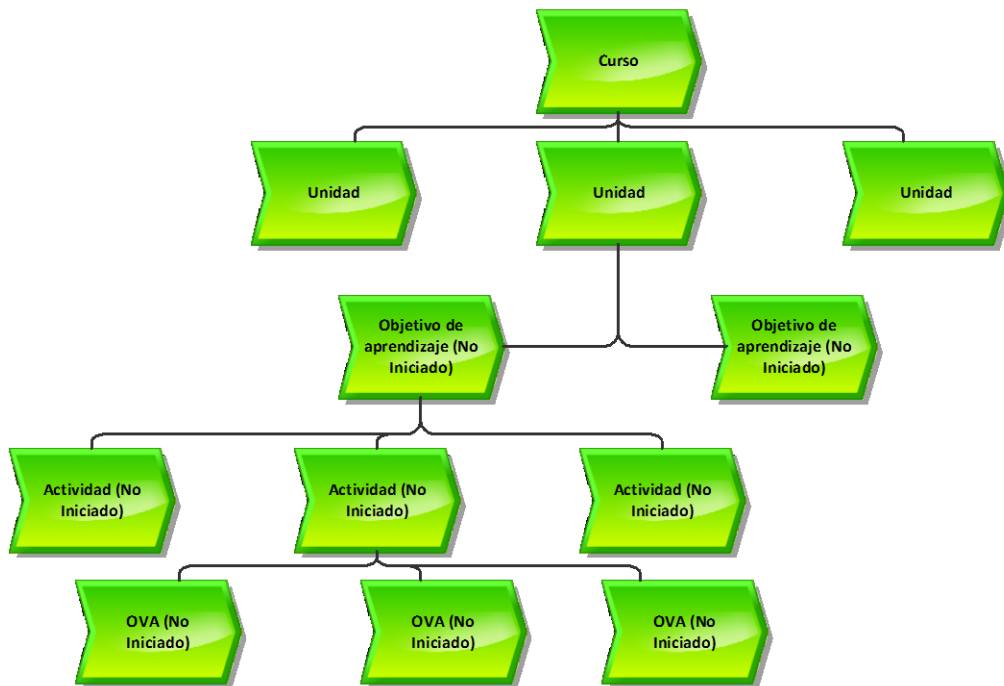


Ilustración 20 Recursos en estado "No Iniciado"

La transición de estados para la ilustración 20 comienza cuando el aprendiz ingresa a desarrollar el OVA que corresponde a la actividad seleccionada del objetivo de aprendizaje seleccionado. Una vez el aprendiz ingresa al OVA comienza la navegación correspondiente para realizar el desarrollo de este (Ver ilustración 18 Modelo de Transición de estado), y el estado de esta jerarquía cambia a “En Proceso”. Esta transición de estado se representa en la siguiente figura.

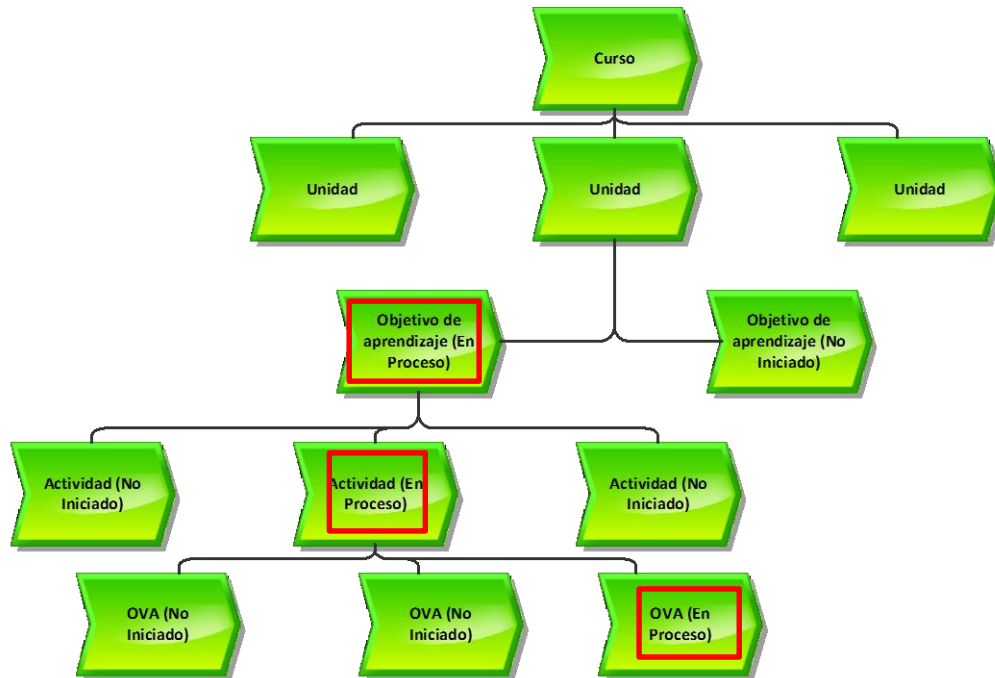


Ilustración 21 Recursos en estado "En Proceso"

Una vez el aprendiz ha culminado el desarrollo del OVA, el estado de este recurso pasa a “Terminado”, pero el estado de los recursos actividad y objetivo de aprendizaje permanece en “En Proceso”, ya que para que se realice la transición de estado los OVAs (para el caso de las actividades) y las actividades (para el caso de las objetivos de aprendizaje) deben estar en estado “Terminado”. La siguiente imagen ilustra lo mencionado anteriormente

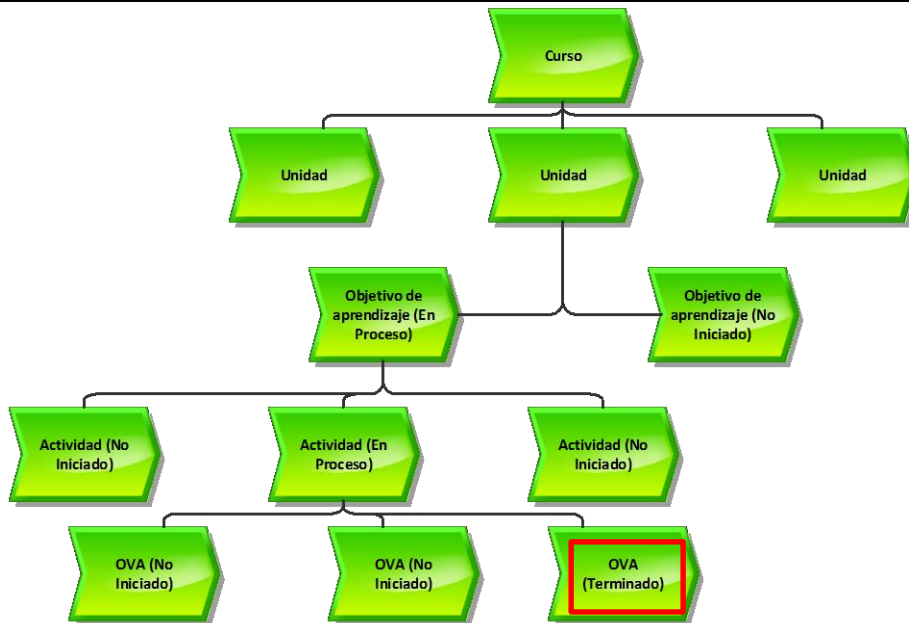


Ilustración 22 Recurso OVA en estado "Terminado"

Una vez que los OVAs y las actividades son culminadas, su estado cambia a "Terminado", por tal motivo el objetivo de aprendizaje cambia su estado al mismo. La siguiente figura ilustra la jerarquía en estado "Terminado".

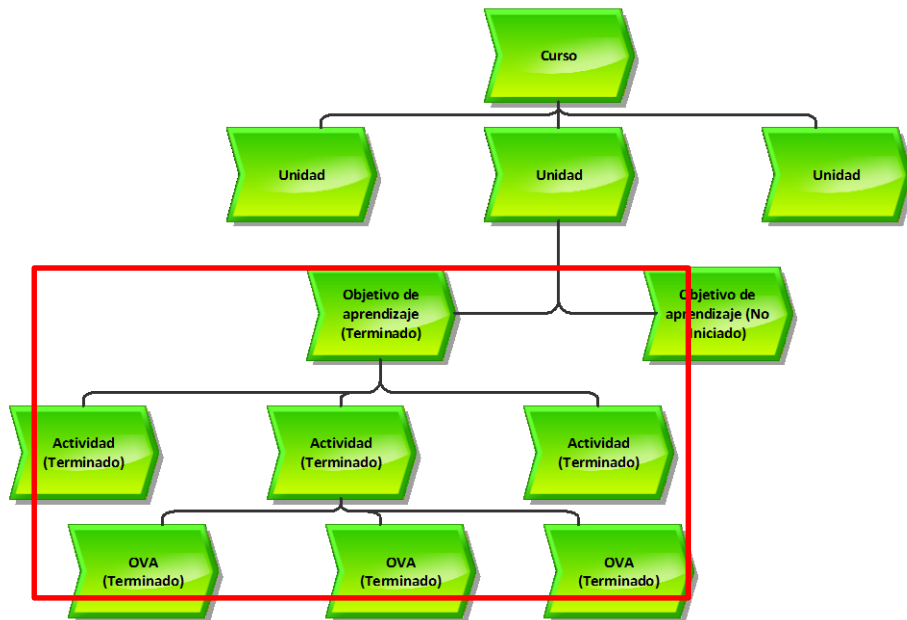


Ilustración 23 Jerarquía con estado "Terminado"

3.3 Diseño de una arquitectura orientada a recursos para e-learning

Luego del planteamiento del protocolo de aplicación y la definición de estados del mismo, en esta sección se procede a realizar el diseño de la arquitectura orientada a recursos que soporte el protocolo definido.

Para la definición de la arquitectura orientada a recursos, se proponen una serie de pasos que permitirán de una manera sistemática definir elementos arquitecturales y relaciones entre sí.

La siguiente imagen muestra el proceso de construcción de la arquitectura.



Ilustración 24 Proceso de construcción de una arquitectura orientada a recursos

A continuación se presentan los pasos correspondientes al diseño metodológico de una arquitectura orientada a recursos:

3.3.1 Definir los elementos arquitecturales siguiendo el estilo REST.

La siguiente figura muestra el overview de la arquitectura orientada a recursos para una aplicación hipertextual en contextos e-learning.

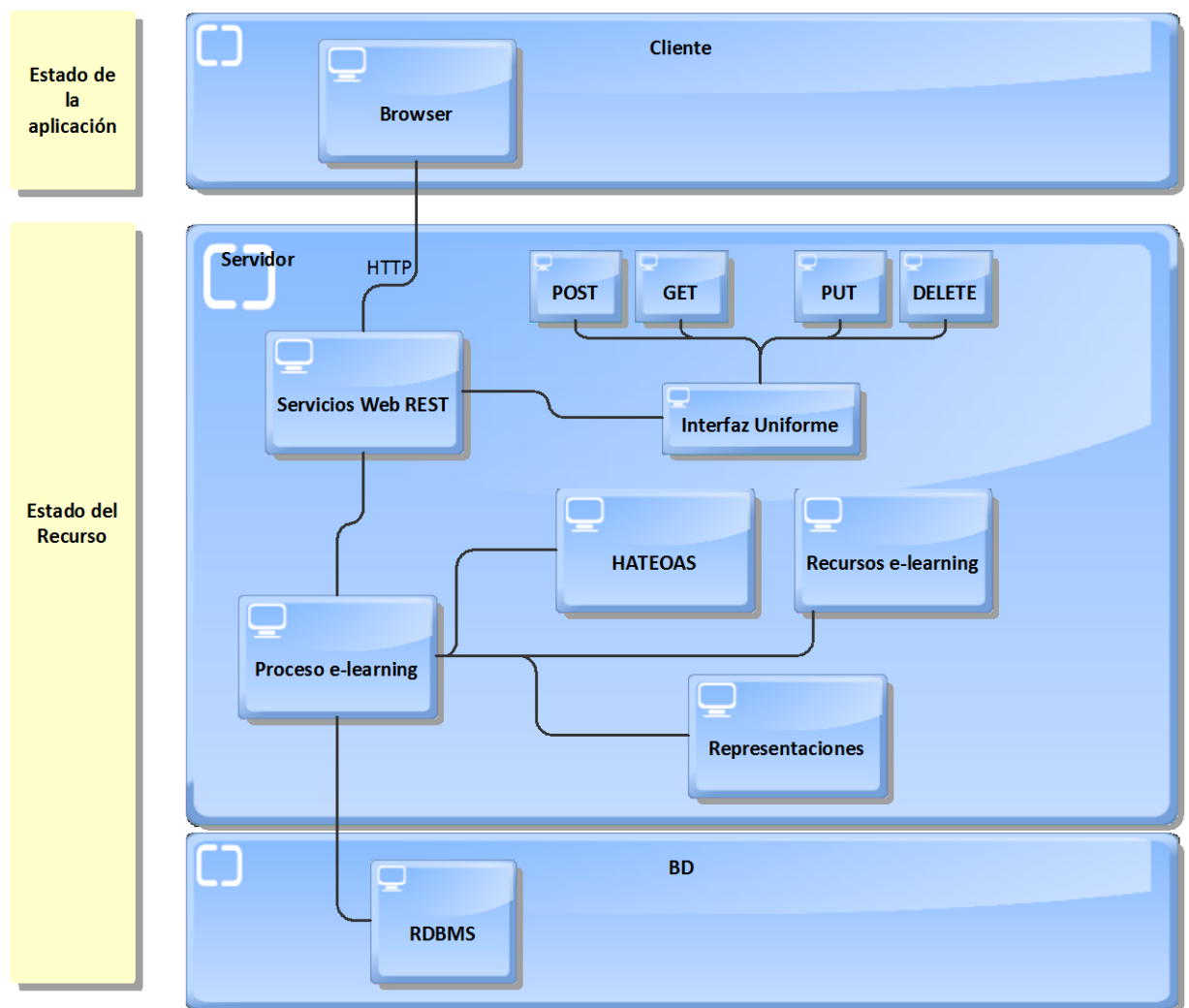


Ilustración 25 OverView de la arquitectura

Los componentes de la figura anterior se describen a continuación:

- Cliente
 - Browser: Componente que permite al usuario interactuar con la aplicación.

- Servidor
 - Servicios Web REST: Componente que expone los recursos por medio de la interfaz uniforme, utilizando métodos HTTP.
 - HETOAS: Componente que contiene los vínculos hipermedia.
 - Proceso e-learning: Componente que contiene la lógica de negocio del proceso de aprendizaje.
 - Representaciones: Componente encargado de generar el formato de los recursos.
 - Recursos e-learning: Componente que contiene los recursos del proceso de aprendizaje, son O.V.A.s que representan los contenidos temáticos.

- BD
 - RDBMS: Componente que representa la persistencia del aplicativo.

La arquitectura propuesta comprende:

- Elementos arquitecturales : Elementos de datos, conectores y componentes
- Vistas: Vista de datos, vista de procesos y vista de conectores.

3.3.1.1 Elementos arquitecturales

Debido a que la arquitectura propuesta se rige por el estilo REST, se requiere obtener una abstracción de los elementos arquitecturales para el sistema distribuido hipermedia planteado (Fielding, UNIVERSITY OF CALIFORNIA, IRVINE, 2000).

3.3.1.2 Elementos de datos

La siguiente tabla define los elementos de datos de la arquitectura propuesta, se muestran dos columnas los elementos de datos (Fielding, UNIVERSITY OF CALIFORNIA, IRVINE, 2000), y sistema e-learning.

Elemento de datos	Sistema e-learning
Recursos	O.V.A., objetivos de aprendizaje, entre otros
Identificador de recursos	URL, URN
Representación	JSON y XML
Representación de metadatos	Media type, last modified time
Metadatos de recursos	Source link
Control de datos	If-modified-since, cache-control

Tabla 5 Elementos de datos y sistema e-learning

- **Recursos e identificadores**

- Recursos

- Objetivos de aprendizaje: Las principales propiedades son: id, nombre, descripción, unidad, objetivo del curso de alto nivel, vínculos a otros objetivos de aprendizaje.
- O.V.A
- Actividades: Las principales propiedades son: id, nombre, descripción, objetivo de aprendizaje, vínculos a otros objetivos de aprendizaje.

- Identificadores de recursos

Se utilizarán URL y URN con la siguiente estructura: curso/unidad/objetivo.

- **Representaciones**

Como se mencionó anteriormente, las representaciones serán JSON y XML, ya que estos proveen un formato capaz de transportar información hipermedia necesaria, y

ofrecen un equilibrio entre utilidad y alcance, así como la habilidad de apalancar agentes existentes (agentes del browser) de software comúnmente usados. Estos formatos concuerdan con las necesidades de representación del dominio e-learning.

- **Representación de metadatos**

Por medio de este elemento, se establecerá la representación deseada del recurso que el servidor debe proveer. Aquí se consideran las siguientes propiedades:

- Media type: application/xml y application/json
- Last modified time: La fecha de última modificación del recurso.

- **Metadatos de recursos**

Se manejará la propiedad Source link, la cual indica los vínculos de los recursos nombrados anteriormente. La estructura del vínculo tiene:

- Link
- Rel
- Href

- **Control de datos**

Se utilizarán controles de datos:

- if-modified-since: si el recurso no es modificado desde una fecha específica, este no será traído desde el servidor sino desde los mecanismos de cache provistos por la arquitectura propuesta.
- cache-control: Es utilizado para verificar si se utiliza o no cache.

3.3.1.3 Conectores

REST usa varios tipos de conectores para encapsular actividades de acceso a los recursos y transferencia de representaciones de estos. Presentan una interfaz abstracta para un componente de comunicación, mejoran la simplicidad proporcionando una clara separación de las inquietudes y ocultando la implementación subyacente de los recursos y el mecanismo de

comunicación (Fielding, UNIVERSITY OF CALIFORNIA, IRVINE, 2000). La siguiente tabla muestra los conectores utilizados para la arquitectura propuesta.

Conectores	Sistema e-learning
Cliente	HTTP
Servidor	JAX-RS
Cache	Browser cache

Tabla 6 Conectores y sistema e-learning

1. Cliente

Inicia la comunicación realizando una petición.

2. Servidor

Escucha a las conexiones y responde a las peticiones con el fin de suplir el acceso a sus servicios.

3. Cache

Puede estar localizado en la interfaz de los conectores cliente y/o servidor, con el fin de guardar respuestas de interacciones actuales para que puedan ser reusadas en interacciones futuras.

3.3.1.4 Componentes

Los componentes hacen todo el procesamiento de datos tal como conversiones y cálculos (Fielding, UNIVERSITY OF CALIFORNIA, IRVINE, 2000). La siguiente tabla muestra los componentes de la arquitectura propuesta.

Componente	Sistema e-learning
Origin server	Glassfish
Gateway	Reverse proxy
User Agent	Mozilla, Internet Explorer

Tabla 7 Componentes y Sistema e-learning

- **Origin server**

Usa el conector servidor para gobernar el namespace de una petición de un recurso. Es la fuente definitiva para las representaciones de sus recursos y debe ser el destino final de cualquier petición que intente modificar el valor de sus recursos.

- **Gateway**

Es un intermediario impuesto por la red o por el origin server que provee una interfaz de otros servicios encapsulados, traducción de datos, mejora del desempeño, o mejora de seguridad.

- **User Agent**

Usa un conector cliente para iniciar una petición y llega a ser el último destino de la respuesta.

En la siguiente figura se muestran los componentes y conectores de la arquitectura lógica de la solución, los conectores que aún no se han definido se aclararán cuando se plantee la arquitectura física.

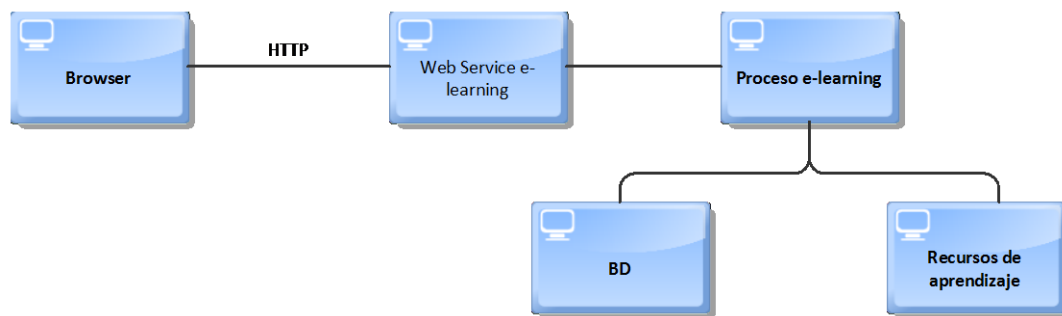


Ilustración 26 Componentes y conectores de la arquitectura lógica

3.3.2 Definir las vistas arquitecturales siguiendo el estilo REST.

3.3.2.1 Vistas arquitecturales

Son usadas para describir como los elementos trabajan juntos para formar una arquitectura. Se utilizaron tres vistas procesos, conectores y datos.

3.3.2.1.1 Vista de procesos

Esta vista se enfoca en el flujo de datos a través de los componentes de la arquitectura y algunos aspectos de las conexiones entre estos, relacionados con los elementos de datos (Fielding, UNIVERSITY OF CALIFORNIA, IRVINE, 2000). Para representar esta vista, se escogieron unas peticiones arquitecturalmente significantes, cuyo detalle se puede ver en el S.A.D (Software Architecture Document, Anexo 5), para ilustrar el contenido de esta vista se usará la petición “Seleccionar un objetivo de aprendizaje”:

- Seleccionar un objetivo de aprendizaje:
 1. El aprendiz selecciona el objetivo de aprendizaje con el cual va a iniciar: Por medio de un browser se realiza una petición al servicio web utilizando una interfaz uniforme de métodos HTTP.
 2. El sistema presenta al aprendiz las actividades correspondientes al objetivo de aprendizaje: el servicio web se conecta con el proceso de e-learning, el cual recupera la información del objetivo de aprendizaje desde la base de datos incluyendo el workflow de actividades correspondientes a este, representando esto en el formato solicitado.
 3. Actualizar el estado del recurso: como se vio en el modelo de transiciones (Ver Sección 3.2.1 Modelo de transiciones entre estados).

La siguiente figura representa el proceso de Selección de un objetivo de aprendizaje explicado anteriormente.

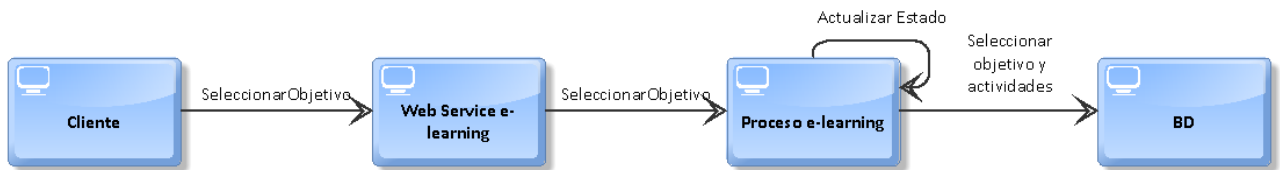


Ilustración 27 Vista de procesos - Seleccionar objetivo de aprendizaje

- Realizar transición:

1. El aprendiz finaliza la última actividad correspondiente al objetivo de aprendizaje, esto es cuando todos los OVAs de la actividad han sido tomados por el aprendiz, se generan los vínculos a los posibles objetivos de aprendizaje que no han sido abordados por el aprendiz de acuerdo al estado del recurso.
2. Actualizar el estado del recurso: A partir de la máquina de estados generada para el escenario de aprendizaje.
3. Actualizar el estado de la aplicación: El aprendiz actualiza el estado de la aplicación cuando selecciona uno de los vínculos posibles presentados en los recursos.

La siguiente figura representa el proceso de Realizar transición explicado anteriormente.

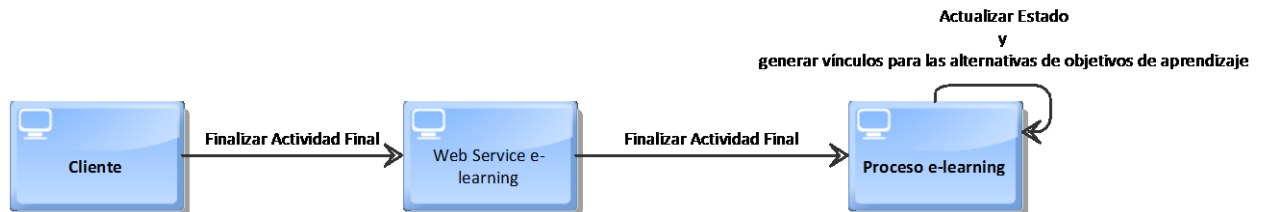


Ilustración 28 Vista de procesos - Realizar transición

3.3.2.1.2 Vista de conectores

Se concentra en los mecanismos de comunicación entre componentes. Para una arquitectura basada en REST, se enfoca en las restricciones que definen una interfaz genérica de recursos (Fielding, UNIVERSITY OF CALIFORNIA, IRVINE, 2000).

Los conectores de clientes examinan el identificador del recurso con el fin de seleccionar el mecanismo de comunicación apropiado para cada petición (Wu & Chang, 2007).

Esta vista será definida cuando se plantee la arquitectura física y se defina la tecnología.

3.3.2.1.3 Vista de datos

Expone el estado de la aplicación mientras que el flujo de información fluye a través de los componentes. El estado de la aplicación es controlado y almacenado por el user agent y puede estar compuesto de representaciones de múltiples servicios (Fielding, UNIVERSITY OF CALIFORNIA, IRVINE, 2000). Por otra parte, se libera al servidor de problemas de escalabilidad almacenando el estado, esto permite al usuario administrar el estado, anticipar cambios del estado, y saltar de una aplicación a otra.

El modelo de aplicación es entonces el motor que se mueve de un estado a otro examinando y escogiendo alternativas de transición de estados en un conjunto de representaciones actuales (Wu & Chang, 2007). Esto concuerda con la interfaz de usuario de un browser hipermedia.

Las máquinas de estados en donde se aprecian los estados (objetivos de aprendizaje, actividades, OVAs) y sus correspondientes transiciones se obtienen a partir del proceso de enseñanza/aprendizaje (Ilustración 18 Modelo de transición entre estados).

Estas transiciones de estado son dinámicas ya que dependen del objetivo de aprendizaje por el cual se inicie.

Para completar la vista de datos, se presenta el siguiente modelo de entidades de negocio para e-learning, basado en los resultados obtenidos en la sección anterior (Ver Sección 3.1.1 Identificando el contenido del curso), en donde se encuentra la descripción de cada uno de ellos.

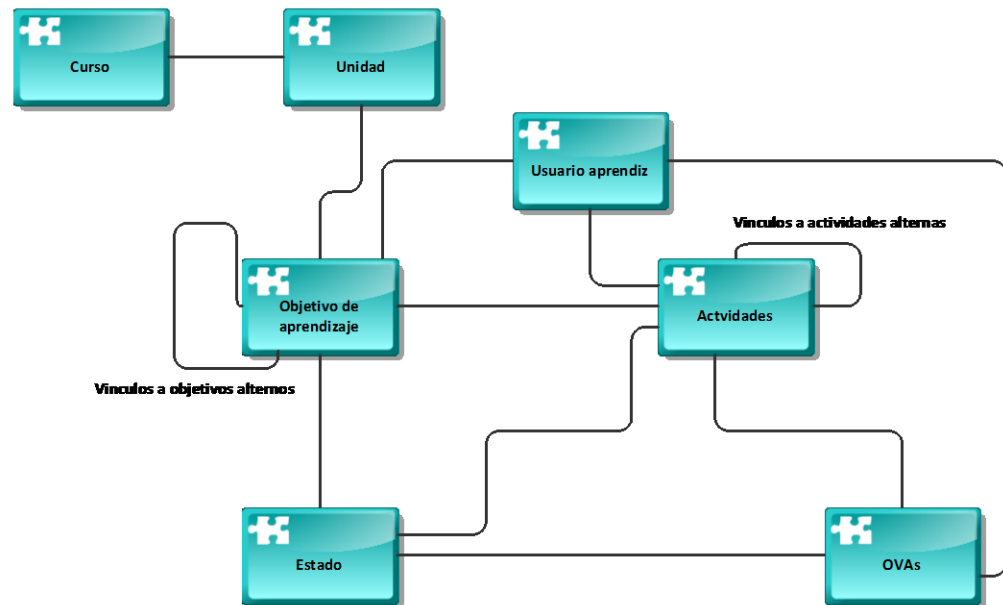


Ilustración 29 Modelo de datos

En la figura anterior, además de las entidades tratadas en esta sección, se incluyen el Estado y el Usuario Aprendiz, las cuales se encargan de mantener el estado de los recursos del sistema teniendo en cuenta la transición de estados presentada en la sección anterior (Ver Sección 3.2.1 Modelo de Transición entre estados).

3.3.3 Diseñar los servicios orientados a recursos.

Una vez se han definido los recursos, los elementos y las vistas arquitecturales, se procede con el diseño de los servicios.

Para realizar el diseño de los servicios, se realizarán los siguientes pasos representados en la siguiente figura.

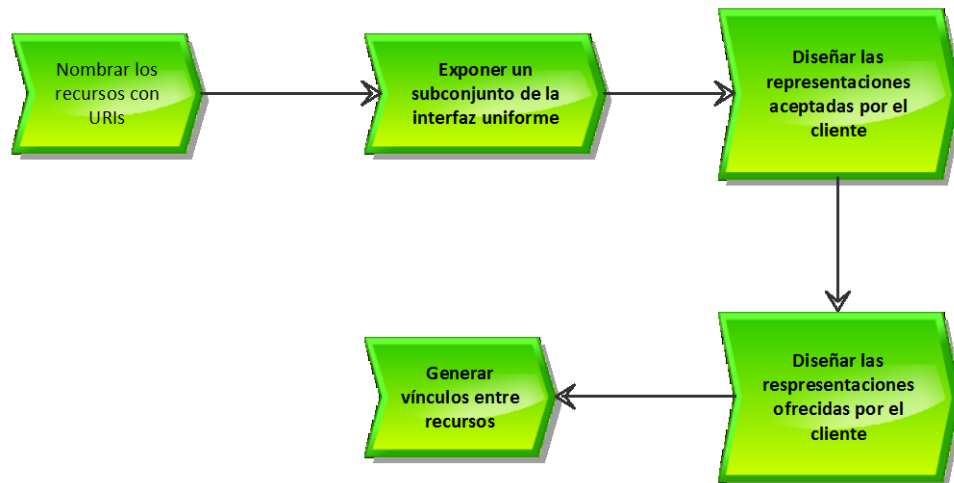


Ilustración 30 Pasos para el diseño de servicios orientados a recursos

3.3.3.1 Nombrar los recursos con URIs

Existen 3 reglas básicas para el diseño de URIs:

- Usar variables path para codificar las jerarquías: /parent/child
- Usar caracteres de puntuación en las variables path para evitar implantar jerarquías donde no las hay: parent/child1;child2
- Usar variables de consulta para informar entradas de un algoritmo: /search?=qjellyfish&start=20

Tomando como referencia la jerarquía presentada en la ilustración 11, la definición de las URIs para los recursos viene dada por variables path para codificar la jerarquía. Algunas de dichas variables path podrían ser:

- /curso/unidad
- /curso/unidad/objetivodeaprendizaje1
- /curso/unidad/objetivodeaprendizaje1/actividad

3.3.3.2 Exponer un subconjunto de la interfaz uniforme

El acceso a los recursos ocurre a través de una interfaz HTTP uniforme (GET, POST, PUT, DELETE, HEAD, OPTIONS) (ruby, 2007).

La siguiente tabla muestra los métodos HTTP:

Método	Operación
POST	Crea un (sub) recurso
GET	Recupera las representaciones de un recurso
PUT	Modifica un recurso /Crea un nuevo recurso
DELETE	Elimina un recurso
OPTIONS	Descubre qué métodos HTTP son soportados por el recurso
HEAD	Únicamente peticiones de cabecera (Similar a GET pero omite representaciones)

Ilustración 31 Métodos HTTP y sus operaciones con recursos

Toda información necesaria para entender la petición debe estar contenida en el mensaje de petición.

La siguiente tabla muestra para cada petición, los recursos que surgen, su respectiva URI, y el método HTTP por el cual es accedido.

Recurso	URI	Método HTTP
Petición: Inicio del curso		
Lista de unidades	/curso/unidades	GET, POST, PUT, DELETE
Petición: Seleccionar una unidad		
Unidad seleccionada	/curso/unidadSeleccionada {IdUnidad}	GET, POST, PUT, DELETE
Lista de objetivos de aprendizaje	/curso/unidadSeleccionada{IdUnidad} /objetivosAprendizaje	GET, POST, PUT, DELETE, HEAD

Petición: Seleccionar un objetivo de aprendizaje		
Objetivo aprendizaje	/curso/unidadSeleccionada{IdUnidad} /objetivoseleccionado{IdObjetivo}	GET, POST, PUT, DELETE, HEAD
Lista de actividades	/curso/unidadSeleccionada {IdUnidad} /objetivoseleccionado{IdObjetivo}/actividades	GET, POST, PUT, DELETE, HEAD
Lista de objetivos alternos	/curso/unidadSeleccionada {IdUnidad} /objetivoseleccionado{IdObjetivo}/objetivosAlternos	GET, POST, PUT, DELETE, HEAD
Petición: Ejecución de las actividades		
Actividad	/curso/unidadSeleccionada {IdUnidad} /objetivoseleccionado{IdObjetivo}/ actividadSeleccionada {IdActividad}	GET, POST, PUT, DELETE, HEAD
OVA's	/curso/unidadSeleccionada {IdUnidad} /objetivoseleccionado{IdObjetivo}/ actividadSeleccionada {IdActividad}/ova {IdOva}	GET, POST, PUT, DELETE, HEAD
Lista de actividades alternas	/curso/unidadSeleccionada {IdUnidad} /objetivoseleccionado{IdObjetivo}/ actividadesAlternas	GET, POST, PUT, DELETE, HEAD
Petición: Realizar transición		
Lista de objetivos alternos	curso/unidadSeleccionada {IdUnidad} /objetivoseleccionado{IdObjetivo}/ actividadSeleccionada {IdActividad}/objetivosAlternos	GET, POST, PUT, DELETE, HEAD

Ilustración 32 Peticiones con sus respectivos recursos, URIs y método HTTP

3.3.3.3 Diseñar las representaciones aceptadas y ofrecidas por el cliente

El siguiente fragmento es una representación en XML de un objetivo de aprendizaje.


```
<?xml version="1.0" encoding="UTF-8"?> version="1.0" encoding="UTF-8" standalone="yes"
  <objetivoa>
    <descripcion>EMR</descripcion>
    <idobjetivo>1</idobjetivo>
    <nombre>Entender modelo relacional</nombre>
    <unidadIdunidad>
      <cursoIdcurso>
        <descripcion>Caso estudio</descripcion>
        <idcurso>1</idcurso>
        <nombre>Bases de Datos</nombre>
      </cursoIdcurso>
      <descripcion>Comprende los conceptos sobre tablas relaciona-
les</descripcion>
      <idunidad>1</idunidad>
      <nombre>Modelo relacional</nombre>
    </unidadIdunidad>
  </objetivoa>
```

El siguiente fragmento es una representación en JSON de un objetivo de aprendizaje.

```
{ "descripcion": "EMR", "idobjetivo": 1, "nombre": "Entender modelo relacio-
nal", "unidadIdunidad": { "cursoIdcurso": { "descripcion": "Caso estu-
dio", "idcurso": 1, "nombre": "Bases de Datos" }, "descripcion": "Comprende los conceptos sobre tablas
relacionales", "idunidad": 1, "nombre": "Modelo relacional" } }
```

La siguiente es la representación para la lista de objetivos de aprendizaje en XML:

```
<objetivosaprendizaje>
  <link href= "http://curso/unidad/objetivo1" name= "objetivo1" />
  < link href= "http://curso/unidad/objetivo2" name= "objetivo2" />
</objetivosaprendizaje>
```

La siguiente es la representación para la lista de actividades en XML:

```
<actividades>
  <link href= "http://curso/unidad/objetivo1/actividad1" name= "actividad1"/>
  < link href= "http://curso/unidad/objetivo1/actividad2" name= "activida2" />
</actividades>
```

La siguiente es la representación para una actividad en XML:

```
<?xml version="1.0" encoding="UTF-8"?> version="1.0" encoding="UTF-8" standalone="yes"
<actividad>
  <descripcion>syllabus</descripcion>
  <idactividad>1</idactividad>
  <nombre>Presentacion curso</nombre>
  <objetivoaIdobjetivo>
    <descripcion>EMR</descripcion>
    <idobjetivo>1</idobjetivo>
    <nombre>Entender modelo relacional</nombre>
    <unidadIdunidad>
      <cursoIdcurso>
        <descripcion>Caso estudio</descripcion>
        <idcurso>1</idcurso>
        <nombre>Bases de Datos</nombre>
      </cursoIdcurso>
      <descripcion>Comprende los conceptos sobre tablas relaciona-
les</descripcion>
      <idunidad>1</idunidad>
      <nombre>Modelo relacional</nombre>
    </unidadIdunidad>
  </objetivoaIdobjetivo>
</actividad>
```

La siguiente es la representación para una actividad en JSON:

```
{ "descripcion": "syllabus", "idactividad": 1, "nombre": "Presentacion curso",
  "objetivoaIdobjetivo": { "descripcion": "EMR", "idobjetivo": 1, "nombre": "Entender modelo rela-
cional", "unidadIdunidad": { "cursoIdcurso": { "descripcion": "Caso estudio", "idcurso": 1, "nombre": "Bases de Datos", "descripcion": "Comprende los conceptos sobre tablas
relacionales", "idunidad": 1, "nombre": "Modelo relacional" } } }
```

3.3.3.4 Generar vínculos entre recursos

Teniendo en cuenta la representación mostrada en el numeral anterior, la siguiente es la representación de un objetivo de aprendizaje con vínculos a otros recursos:

```
<objetivoaprendizaje>
  <id>1</id>
  <nombre>Objetivo de aprendizaje 1</nombre>
  <descripción>descripción del objetivo de aprendizaje</descripción>
  <unidad>temática 1</unidad>
  <objetivoAltonivel>objetivo</objetivoAltonivel>
  <vínculos>
    <link rel="" href: http://curso/unidad/objetivo2/>
    < link rel="" href: http://curso/unidad/objetivo3/>
  </vínculos>
</objetivoaprendizaje>
```

La siguiente es la representación de una actividad con vínculos a otros recursos:

```
<actividad>
  <id>1</id>
  <nombre>Actividad 1</nombre>
  <descripción>descripción de la actividad</descripción>
  <unidad>temática 1</unidad>
  <objetivoAltonivel>objetivo</objetivoAltonivel>
  <objetivoaprendizaje>objetivoaprendizaje1 </objetivoaprendizaje>
  <vínculos>
    <link rel="" href: http://curso/unidad/objetivo1/actividad1/>
  </vínculos>
</ actividad >
```

Para finalizar el diseño de los servicios, la siguiente ilustración muestra la distribución de los componentes a través de nodos.

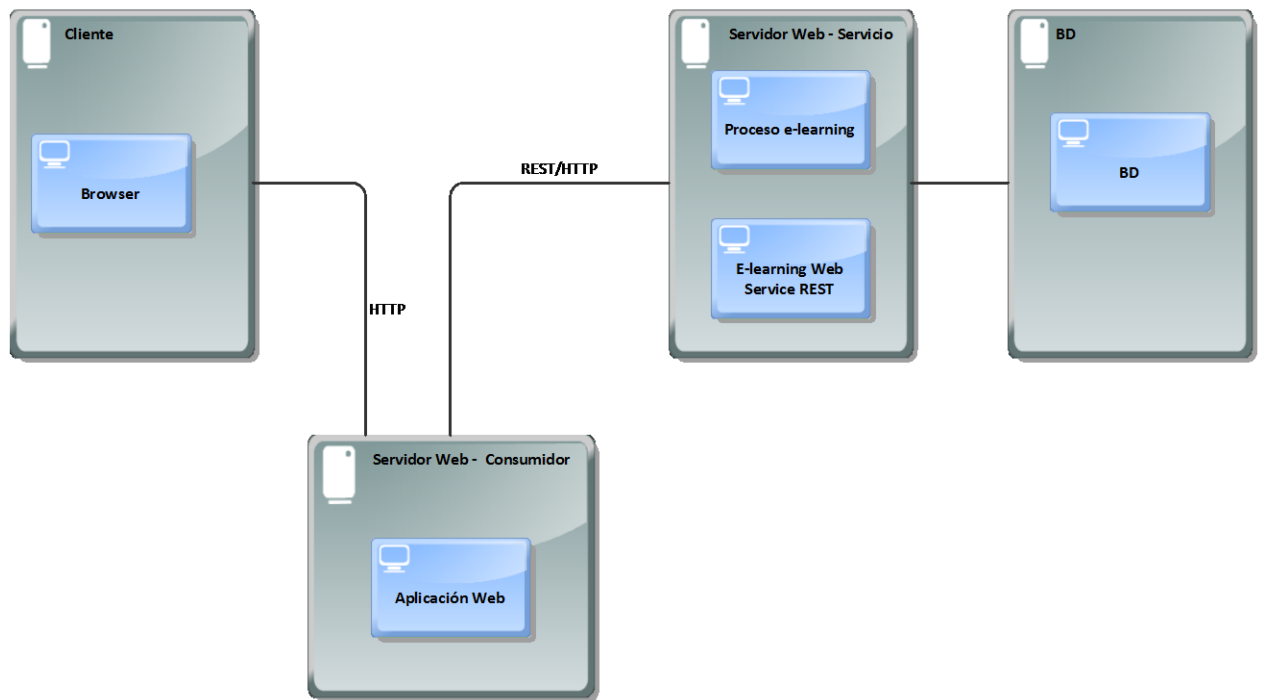


Ilustración 33 Distribución de componentes en nodos

En la ilustración anterior se pueden distinguir los siguientes componentes:

- Nodo Cliente: contiene el componente browser que permite visualizar el aplicativo.
- Nodo servidor web – consumidor: contiene el componente aplicación web, el cual consume el servicio.
- Nodo servidor web – servicio: contiene los componentes proceso e-learning el cual fue definido en la sección anterior (Ver Sección ## definición proceso e-learning), y E-learning web service REST que provee el servicio a consumir.
- Nodo BD: contiene el componente BD que representa la base de datos del sistema.

IV – PROTOTIPO DE APLICATIVO ROA PARA E-LEARNING

En esta sección se instanciará la arquitectura lógica planteada de acuerdo a una tecnología específica (lo que se llamará arquitectura física). El proceso consiste en los mismos pasos planteados para la construcción de la arquitectura física pero teniendo en cuenta un curso específico de la carrera de ingeniería de sistemas y una tecnología específica como es Java EE 7.0

4.1 Proceso de construcción de un curso de e-learning

Para la construcción del curso que será parte del aplicativo, se tomará el modelo mencionado en la sección anterior (Ver Sección 3.1 Proceso de construcción de un curso de e-learning), permitiendo definir las actividades que se desarrollarán en este.

4.1.1 Análisis

En esta fase se identifican los contenidos del curso, los cuales dan lugar al objetivo de alto nivel. A continuación se presenta el desarrollo de esta fase basada en el análisis de tópicos (Ver Sección 3.1.1.1 Identificando el contenido del curso).

4.1.1.1 Identificando el contenido del curso

Teniendo en cuenta el análisis de tópicos ya explicado con anterioridad, se identificó y clasificó el contenido del curso. Para realizar éste proceso, en éste trabajo de grado se tomó el syllabus de la asignatura Bases de Datos, ya que brinda las categorías de contenido y sus detalles. La siguiente imagen ilustra el objetivo principal de la asignatura y el contenido asociado a esta, para una mayor claridad los temas fueron agrupados por unidades.

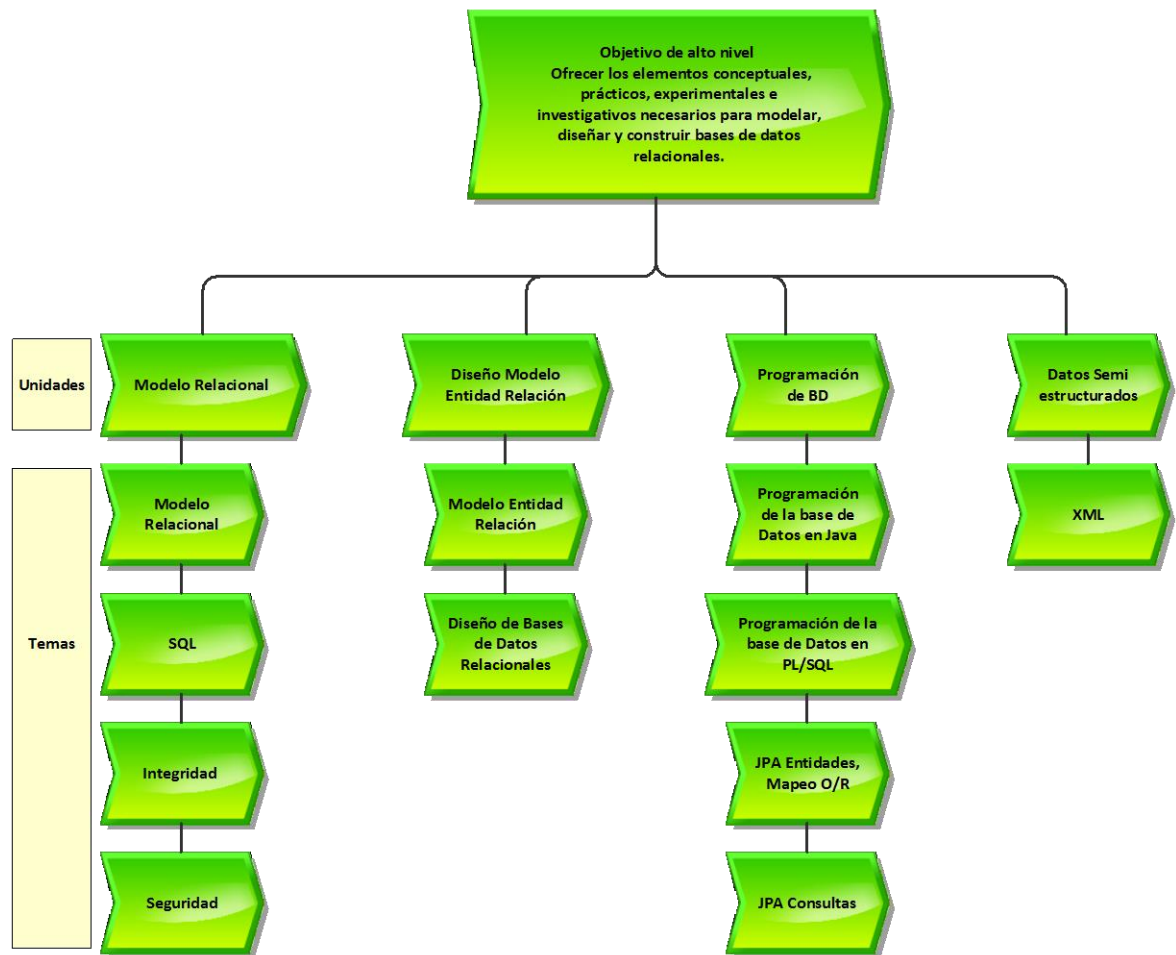


Ilustración 34 Contenido del curso bases de datos

La clasificación dada a los elementos de contenido, se realizó de acuerdo a los tipos de contenidos que ellos representan. Teniendo en cuenta la clasificación presentada en la sección anterior (Ver Sección 3.1.1.1 Identificando el contenido del curso), la clasificación que se muestra en la figura 34 tiene tipos de elementos de contenido tales como hechos, procedimientos, conceptos y principios, a continuación se presenta una tabla de tipos de elementos de contenido y elementos de contenido del curso de Bases de Datos.

Tipos de elementos de contenido	
Hechos	Integridad

Procedimientos	SQL, XML, Programación de bases de datos en java, Programación de bases de datos PL/SQL, JPA consultas
Conceptos	Modelo relacional, Modelo entidad relación, seguridad
Principios	Diseño de bases de datos relacionales, JPA entidades mapeo O/R

Tabla 8 Tipos de elementos de contenido y elementos de contenido para la asignatura bases de datos

4.1.2 Diseño

Teniendo en cuenta la identificación y clasificación de contenido presentada en el numeral anterior, esta fase de diseño tiene como función principal formular los objetivos de aprendizaje para la asignatura de bases de datos y el orden en que deben ser alcanzados. Teniendo en cuenta los niveles de rendimiento para el dominio cognitivo presentados en la sección anterior (Ver Sección 3.1.2.1 Definición de objetivos de aprendizaje), la siguiente tabla lista los objetivos de aprendizaje formulados para la asignatura de bases de datos.

Objetivos de aprendizaje para la asignatura Bases de Datos
1. Entender el modelo relacional
2. Aplicar el modelo relacional
3. Entender seguridad
4. Aplicar seguridad
5. Entender SQL
6. Aplicar SQL
7. Entender integridad
8. Aplicar integridad
9. Entender el diseño del modelo entidad relación
10. Analizar el diseño del modelo entidad relación
11. Aplicar el diseño del modelo entidad relación
12. Entender el modelo entidad relación
13. Analizar el modelo entidad relación
14. Aplicar el modelo entidad relación

15. Entender el diseño de bases de datos relacionales
16. Analizar el diseño de bases de datos relacionales
17. Aplicar programación de base de datos
18. Entender entidades JPA mapeo O/R
19. Entender consultas JPA
20. Aplicar consultas JPA
21. Entender datos semiestructurados
22. Entender XML

Tabla 9 Objetivos de aprendizaje para la asignatura bases de datos

Una vez se encuentran definidos los objetivos de aprendizaje como se muestra en la Tabla 9, se procede a definir las actividades correspondientes para cada objetivo de aprendizaje siguiendo lo estipulado en la sección anterior (Ver Sección 3.1.2.2 Definición de Actividades), la siguiente tabla muestra los componentes requeridos para este caso de estudio, junto con la estrategia de aprendizaje adecuada y su correspondiente objeto virtual de aprendizaje.

Componente	Estrategia de aprendizaje	OVA
General	Acceso a la información	Syllabus (HTML)
Eventos de aprendizaje	Acceso a la información	Enunciado laboratorios prácticos (HTML), seguimiento de tutoriales (HTML)
Comunicación	Acceso a la información	Presentaciones ppt (HTML)
Proyectos grupales	Acceso a la información	Enunciado proyecto (HTML)
Auto Estudio	Acceso a la información	Enunciado talleres (HTML)

Tabla 10 Actividades para la asignatura bases de datos

Ya que debe existir una alineación entre lo presentado en las tablas 9 y 10. Una vez definidas las actividades entonces para cada objetivo de aprendizaje se instancia el conjunto de actividades correspondiente. La definición de actividades para cada objetivo de aprendizaje se

obtiene a partir del experto del curso, en este caso el tutor del trabajo de grado quien es el profesor de la materia de bases de datos.

La siguiente imagen muestra las Actividades para objetivos de aprendizaje de la unidad Modelo Relacional. (Las demás definiciones se encuentran en el Anexo 8. del diseño del curso de e-learning)

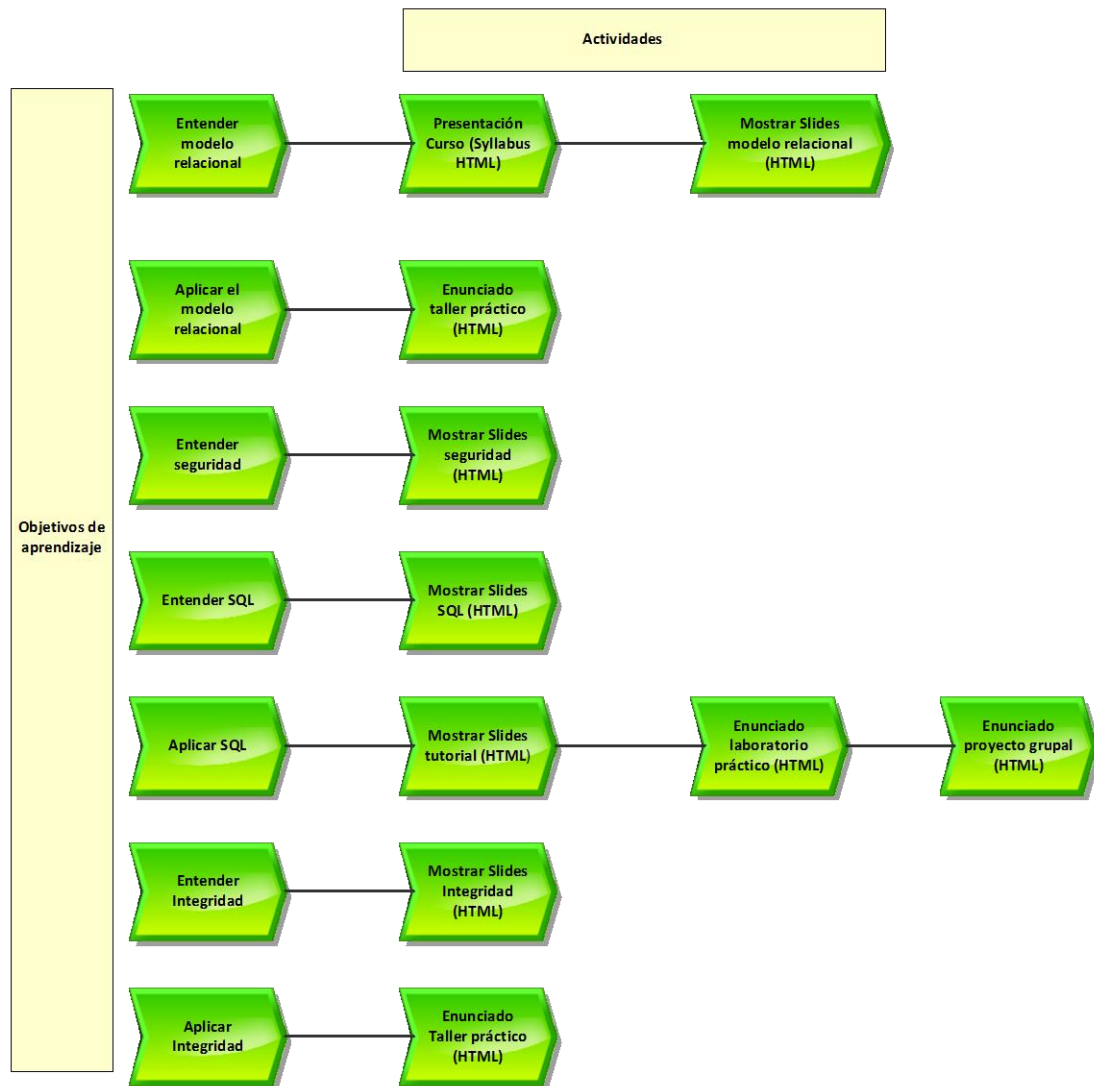


Ilustración 35 Actividades para objetivos de aprendizaje de la unidad modelo relacional

4.1.3 Desarrollo

En esta fase se produce el contenido que será presentado en el curso, tal como se planteó en la sección anterior (Ver Sección 3.1.3 Desarrollo). Para este caso de estudio, los contenidos serán archivos en formato PDF y .ppt exportados a HTML para su visualización en el browser. El desarrollo de este contenido se basó 2 pasos:

- Desarrollo de contenido: Como se mencionó en la sección 3.1.3, este paso es la recolección de todo el contenido o información requerida. Para este paso se tomó como base el syllabus de la asignatura base de datos, con el fin de recolectar información suficiente para estructurar el curso. Por otra parte este paso fue apoyado por los libros guías de la asignatura:
 - Fundamentos de diseño de bases de datos 6a ed. en inglés Silberschatz, Abraham. 2011.
 - SILBERSCHATZ, ABRAHAM, et al. Fundamentos de Bases de Datos. Quinta Edición. McGraw Hill. 2005.
- Desarrollo del storyboard: Este paso integra métodos instruccionales y los elementos mediales utilizados para el desarrollo del curso. Para el caso de estudio, los elementos mediales son Objetos Virtuales de Aprendizaje (OVA) construidos en HTML. Cada OVA consiste de un visor de imágenes (cada imagen es un slide del contenido del curso) y vínculos de navegación entre los diferentes slides. Los vínculos de navegación (permuten cambiar el estado de la aplicación) corresponden a: ir al primer slide, ir al siguiente slide, ir al anterior slide y el slide de Terminar (que permite cambiar el estado del recurso) Para ver un detalle de los story board del caso de estudio refiérase al Anexo 6.

4.2 Definición del protocolo de aplicación en e-learning

Esta fase hace parte de la sección de diseño presentada en el modelo ADDIE (Ver Sección 3.3 Definición del protocolo de aplicación en e-learning), en donde una vez definidos los objetivos de aprendizaje y las actividades respectivas, se plantea el protocolo de aplicación que indica como estos elementos deben ser secuenciados. Teniendo en cuenta el método de

pre requisitos, a continuación se presenta una imagen en donde se aprecia una jerarquía de objetivos de aprendizaje basada en los resultados del análisis de tópicos (Ver Sección 4.1.1 Análisis).

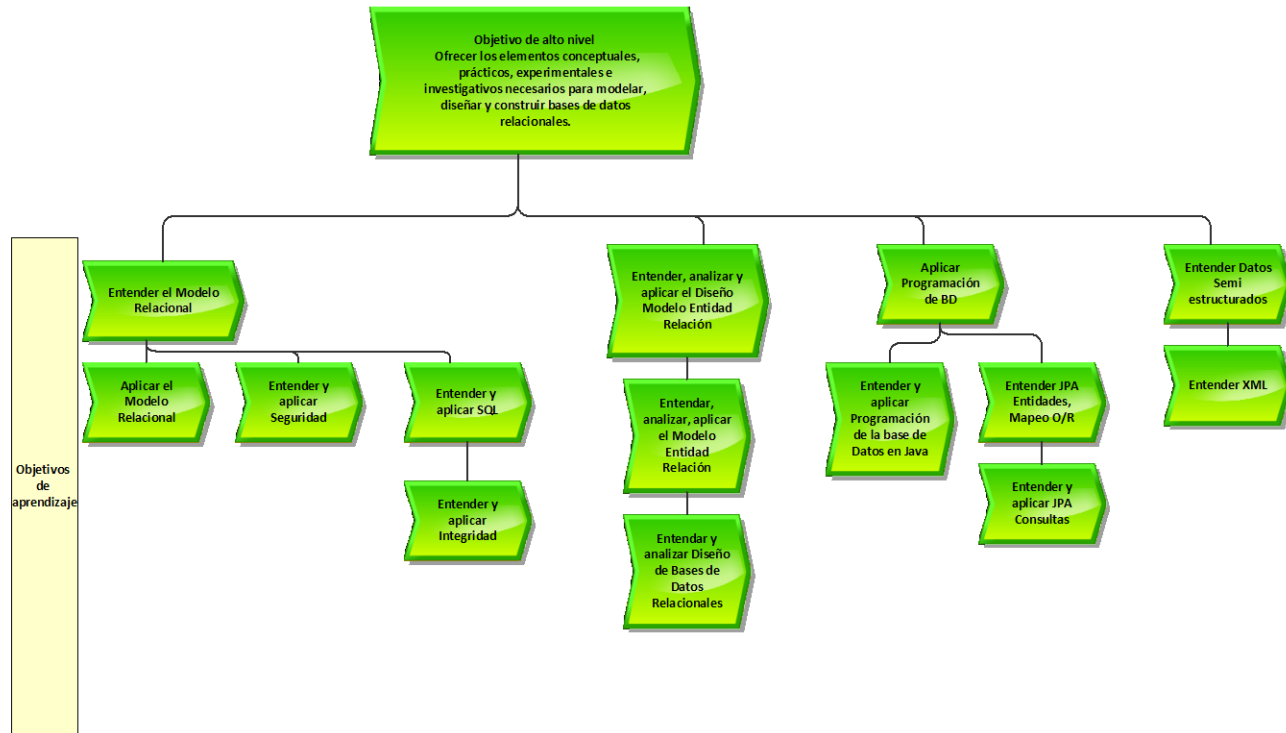


Ilustración 36 Jerarquía de objetivos de aprendizaje

Como se ilustra en la ilustración 36, los objetivos de aprendizaje fueron jerarquizados de tal manera que es fácil apreciar la estructura del curso, como se explicó en la sección anterior (Ver Sección 3.3 Definición del protocolo de aplicación en e-learning), los objetivos que se encuentran en el mismo nivel son independientes entre ellos, para éste caso los objetivos de aprendizaje independientes son:

- Entender el modelo relacional.
- Entender, analizar y aplicar el diseño del modelo entidad relación.
- Aplicar programación de bases de datos.
- Entender datos semi estructurados.

Una vez están establecidos los objetos de aprendizaje independientes, el aprendiz puede determinar por cual rama empezar su proceso de aprendizaje, de igual modo éste manejará el estado de los recursos durante dicho proceso.

A continuación se presenta una posible máquina de estados generada para un solo caso a partir de la ilustración 36.

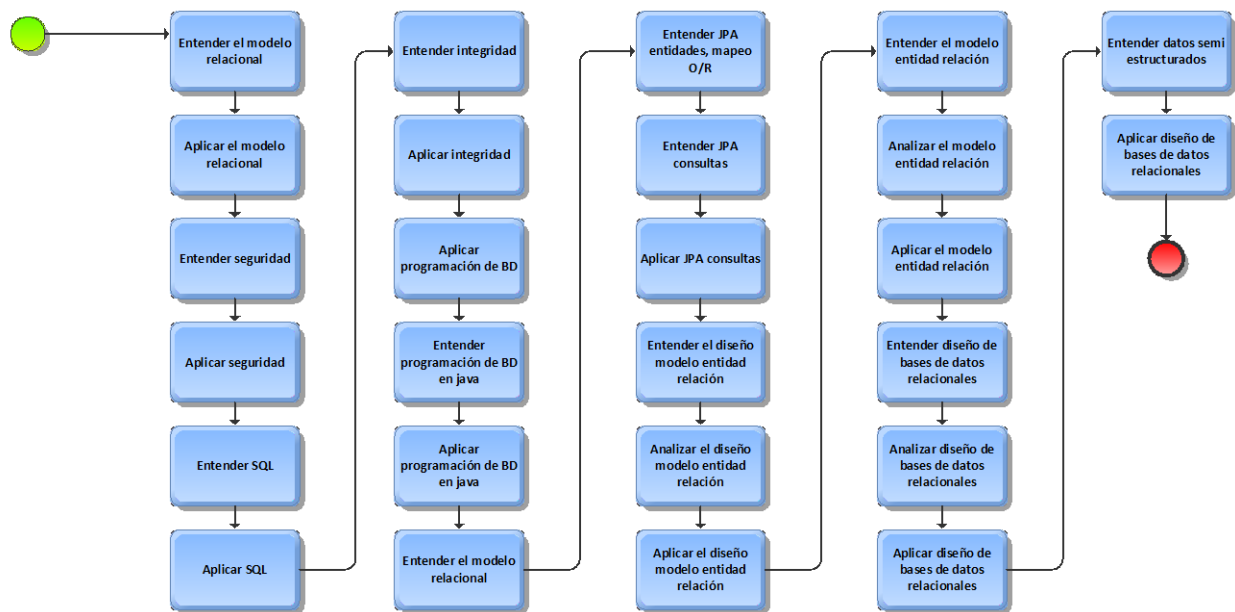


Ilustración 37 Máquina de estados para objetivos de aprendizaje de la asignatura bases de datos

Ya que en la sección anterior se determinó que las actividades correspondientes a cada objetivo de aprendizaje también contarían con el mismo protocolo de aplicación de los objetivos de aprendizaje (Ver sección 3.3 Definición del protocolo de aplicación en e-learning), a continuación se presentan unas imágenes con la jerarquía de actividades para algunos de los objetivos de aprendizaje.

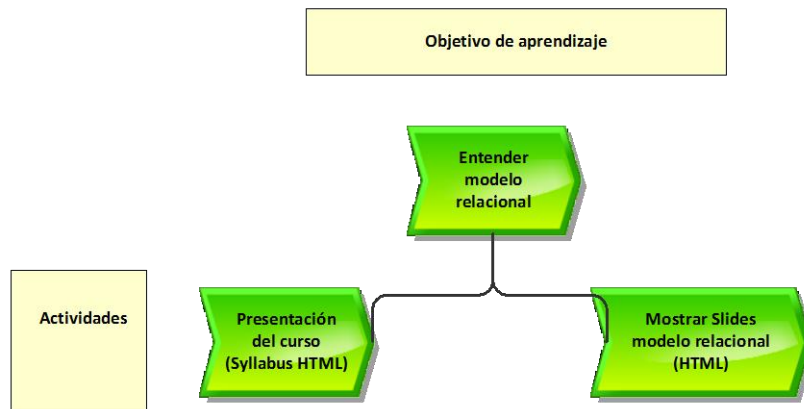


Ilustración 38 Jerarquía actividades para el objetivo de aprendizaje "Entender modelo relacional"

La ilustración 38 muestra dos actividades para un solo objetivo de aprendizaje, al igual que la jerarquización presentada anteriormente para los objetivos de aprendizaje, las actividades que se encuentran al mismo nivel son independientes entre ellas, por lo tanto no son pre requisito una de la otra.

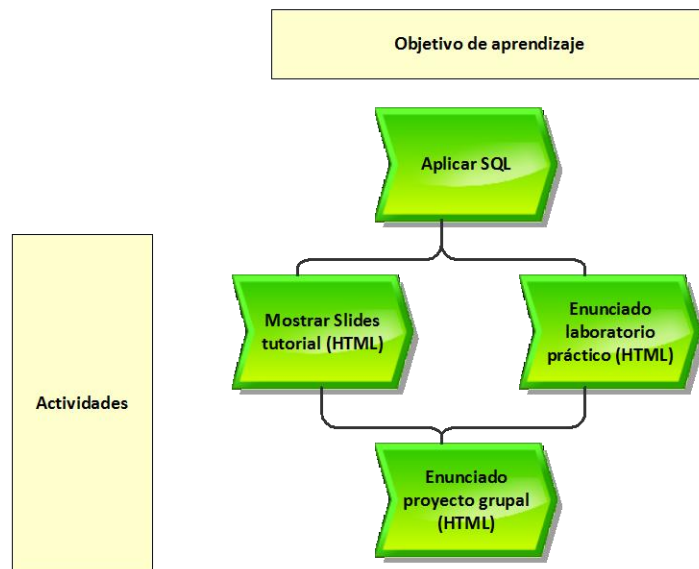


Ilustración 39 Jerarquía actividades para el objetivo de aprendizaje "Aplicar SQL"

A diferencia de la ilustración 38, la ilustración 39 muestra dos actividades independientes y una dependiente, por lo tanto las actividades de mostrar slides tutorial (HTML) y enunciado laboratorio práctico (HTML) son independientes entre sí pero son pre requisito para la actividad enunciado proyecto grupal (HTML).

Tal como se ilustró en las ilustraciones 38 y 39, ocurre lo mismo con el resto de objetivos de aprendizaje y sus actividades, las máquinas de estados que se generarían también dependen de cómo el aprendiz guía su propio proceso de aprendizaje.

4.3 Diseño de una arquitectura orientada a recursos para e-learning

Luego del planteamiento del protocolo de aplicación para el caso de estudio en el numeral anterior, se procede a realizar el diseño de la arquitectura orientada a recursos continuando con los pasos propuestos en la sección anterior (Ver Sección 3.5 Diseño de una arquitectura orientada a recursos para e-learning), en donde se definieron los elementos arquitecturales, conectores y vistas.

A continuación se presenta la arquitectura física aterrizada al caso de estudio y tomando como base la arquitectura lógica presentada en la sección anterior.

4.3.1 Diagrama de Componentes

La siguiente imagen muestra el diagrama de clases en donde se muestran los diferentes elementos arquitecturales para el caso de estudio, componentes y conexiones entre ellos.

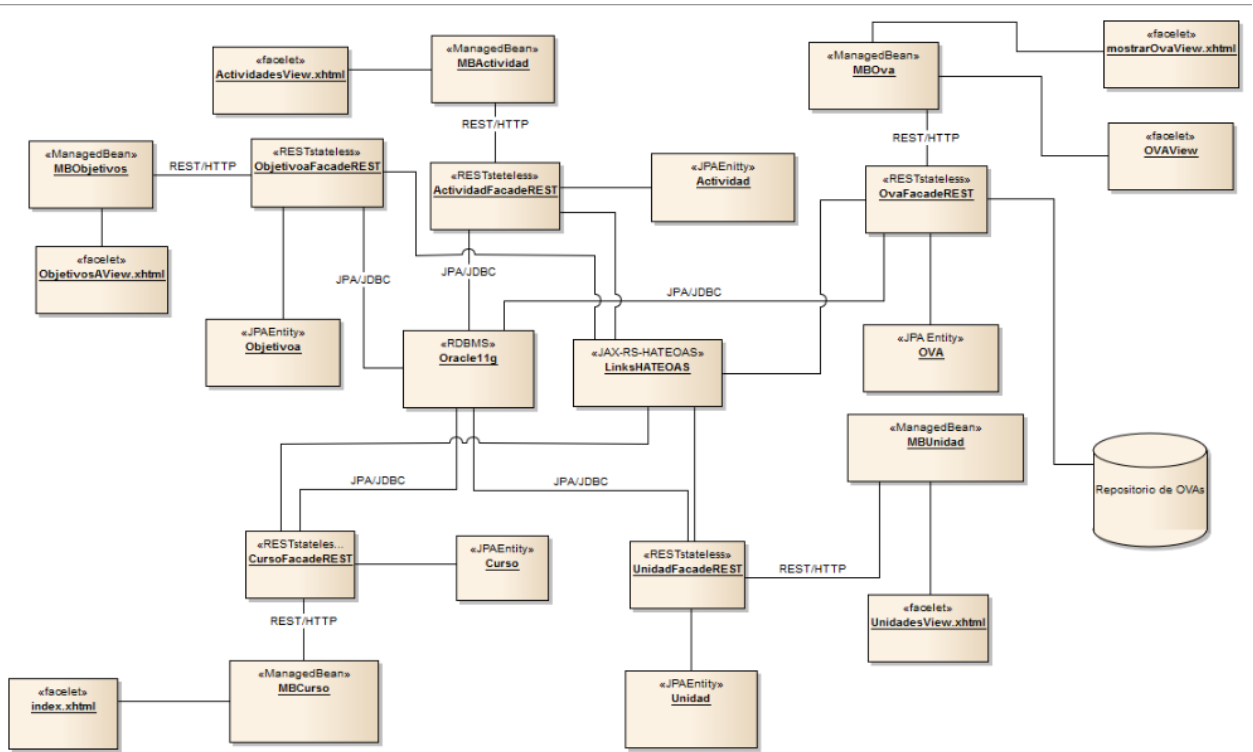


Ilustración 40 Diagramas de componentes caso de estudio

En el anterior diagrama los componentes marcados con estereotipo:

- **RESTStateless:** contienen la lógica de negocio asociado al recurso (lo expone como un servicio).
- **JPAEntity:** Entidades que contienen datos persistentes de los recursos.
- **ManagedBean:** Controladores para los consumidores web encargados de consumir el servicio expuesto por el RESTStateless.
- **Facet:** Componente JSF para la vista de los consumidores.
- **RDBMS:** Componente que representa la base de datos relacional donde se guardan los recursos de manera persistente.
- **Repositorio OVAs:** Directorio en el sistema de archivos que guarda la representaciones en imágenes y HTML de los recursos.
- **JAX-RS HATEOAS:** Componente que permite crear los links que los recursos.

4.3.2 Diagramas de secuencia

Se utilizaron diagramas de secuencia para mostrar la interacción entre los elementos representados en el numeral anterior, indicando los métodos necesarios para completar cada una de las peticiones arquitecturalmente significantes definidas en la sección anterior, el detalle completo de los diagramas se encuentra en el S.A.D. (Software Architecture Document). Para ilustrar la interacción de componentes se utilizará la petición “Listar Objetivos de la Unidad”

4.3.2.1 Diagrama de secuencia Listar Objetivos de la Unidad

La siguiente imagen muestra el secuenciamiento requerido para listar los objetivos de aprendizaje de una unidad seleccionada:

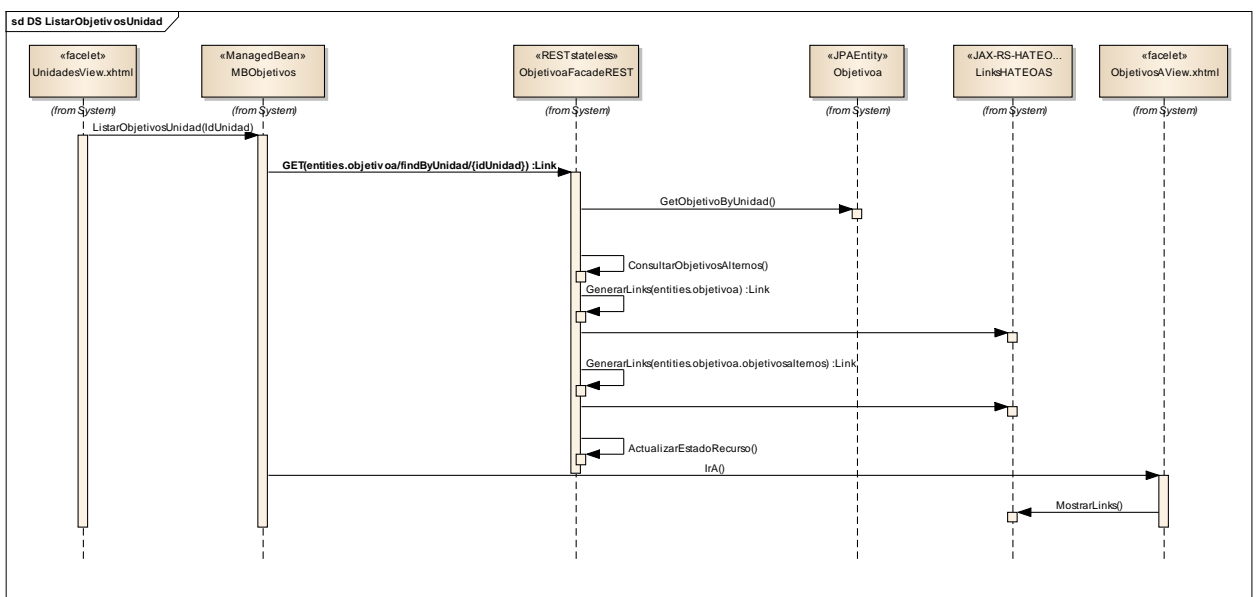


Ilustración 41 Diagrama de secuencia para listar objetivos de aprendizaje de una unidad

En el diagrama de secuencia representado en la ilustración 41, interactúan los siguientes elementos como se describe a continuación:

1. UnidadesView.xhtml solicita a MBOjetivos una lista de objetivos de aprendizaje dada una unidad específica.
2. MBOjetivos solicita por medio de un método GET al ObjetoaFacadeREST, los links correspondientes a los objetivos de aprendizaje de la unidad solicitada.
3. ObjetoaFacadeREST solicita al elemento de persistencia la lista de objetivos de aprendizaje según la unidad seleccionada.

4. Ya que algunos objetivos de aprendizaje son dependientes de otros, el `ObjetivoFacadeREST` primero realiza esta consulta y posteriormente generar los links pertinentes de estos objetivos de aprendizaje.
5. Una vez generados los link de los objetivos dependientes, el `ObjetivoFacadeREST` se comunica con links con el fin de crear los vínculos.
6. Una vez culminado el paso anterior, el `ObjetivoFacadeREST` genera los links de los objetivos que son independientes.
7. El `ObjetivoFacadeREST` repite el paso 5 para los objetivos de aprendizaje independientes.
8. Una vez se encuentran los links generados, el `ObjetivoFacadeREST` actualiza el estado de este recurso.
9. Con los links de los objetivos de aprendizaje ya definidos, `MBObjetivo` procede a comunicarse con `ObjetivosView.xhtml` quien a su vez se comunica con links y muestra en pantalla los links de los objetivos de aprendizaje, indicando cuales son independientes y cuáles no.

4.3.3 Vista de datos

En la vista de datos de la arquitectura propuesta se encuentran las máquinas de estados de los recursos principales del sistema: objetivos de aprendizaje, actividades y objetos virtuales de aprendizaje. Allí también se encuentran los modelos de transición de estados y el modelo de datos que permite persistir el estado de los recursos.

En esta sección se muestra una máquina de estado general del curso de e-learning, en donde se aprecia la interacción entre las principales entidades del sistema.

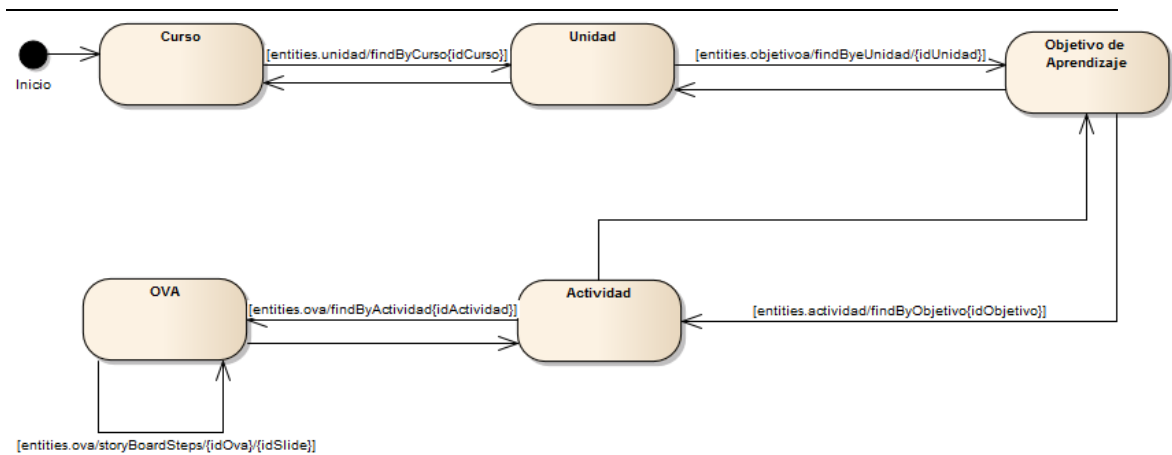


Ilustración 42 Máquina de estados del sistema

La figura anterior presenta el proceso general del sistema, en donde se listan los diferentes elementos que posteriormente siguen los procesos establecidos por los diagramas de secuencia.

4.3.4 Vista de implementación

Una vez establecidos el diagrama de clases, los diagramas de secuencia y la máquina de estados general del sistema, se procede a presentar los diagramas de secuencia correspondientes, en donde se incluyen los elementos de datos identificados previamente.

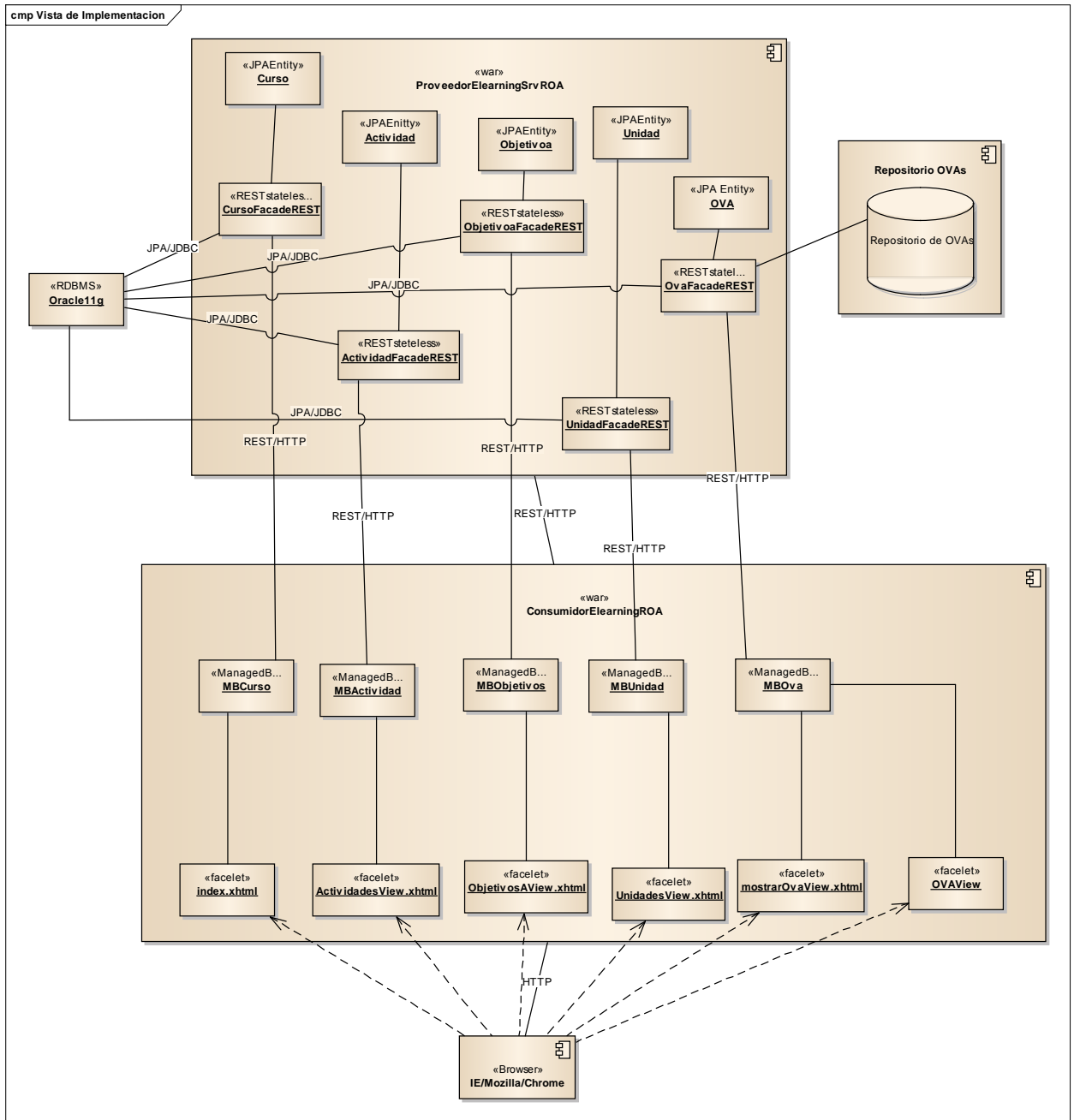


Ilustración 43 Diagrama de componentes del sistema

En la figura anterior se aprecian dos componentes principales, uno de ellos es el servicio (ProveedorElearningSrvROA con estereotipo war) en donde se encuentran aquellos ele-

mentos con estereotipo JPAEntity y RESTSTATELES, ya que estos están encargados de la lógica de negocio y de proveer el servicio que será consumido por el cliente; el segundo componente que se puede apreciar en la figura es el cliente (ConsumidorElearningROA con estereotipo war) el cual contiene aquellos elementos con estereotipo ManagedBean y facet, los cuales se comunican con los elementos de servicio, consumiendo su funcionalidad.

Por otra parte también se pueden apreciar otros componentes que interactúan dentro del sistema:

- IE/MOZILLA/CHROME (estereotipo Browser): este componente interactúa directamente con los elementos del cliente que son de estereotipo facet.
- Oracle11g (estereotipo RDBMS): este componente interactúa directamente con los elementos del servicio que tienen estereotipo RESTSTATELESS.
- Repositorio OVAs: este componente interactúa únicamente con el elemento de servicio llamado OvaFacadeRest de estereotipo RESTSTATELESS.

4.3.5 Vista de capas

Ya que en el anterior numeral se presentó la viste de implementación con cada uno de los componentes que hacen parte del sistema, a continuación se muestra una vista de más alto nivel, en donde se aprecian las capas del sistema.

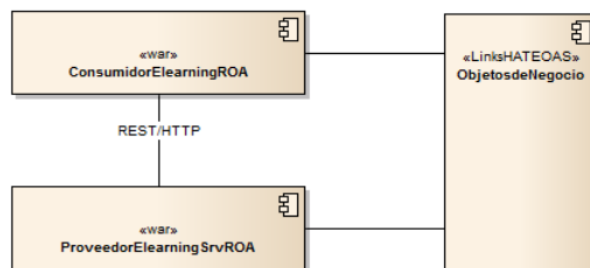


Ilustración 44 Diagrama de capas del sistema

En la figura anterior se aprecian las capas del servicio y el consumidor del sistema, ambas con estereotipo war y se comunican entre ellas por medio de una conexión REST/HTTP, por otra parte hay una capa transversal que contiene los objetos de negocio y que es de es-

tereotipo LINKSHATEOAS, se comunica con las capas de servicio y consumidor ya que ambos utilizan el intercambio de links entre recursos del sistema.

4.3.6 Vista física

Una vez definidos los componentes como se mostró en los numerales anteriores, se procede a ubicar dichos componentes en nodos para representar la distribución del sistema en diferentes máquinas. A continuación se presenta una imagen de la vista física del sistema.

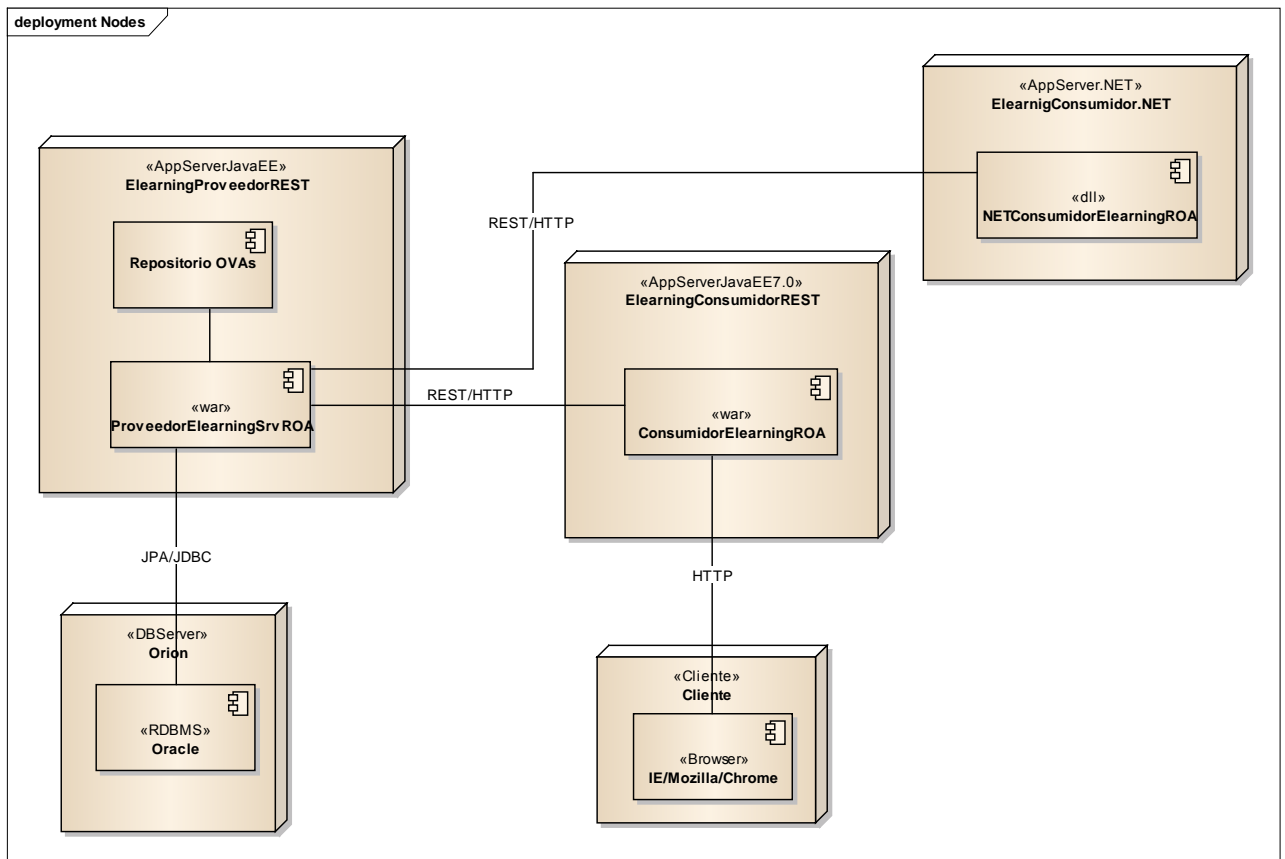


Ilustración 45 Vista física del sistema

La figura anterior ilustra la distribución del sistema en nodos diferentes:

- ElearningProveedorREST: contiene los componentes de repositorio OVAs y ProveedorElearningSrvROA, explicados anteriormente.

-
- Orion: Este nodo representa el servidor de base de datos en donde se encuentra la persistencia de datos del sistema. Se conecta con el proveedor del servicio por medio de una conexión tipo JPA/JDBC.
 - ElearningConsumidorREST: contiene los componentes de ConsumidorElearningSrvROA explicado con anterioridad. Se comunica con uno de los componentes del proveedor del servicio por medio de una conexión REST/HTTP.
 - Cliente: contiene el componente IE/MOZILLA/CHROME que representa el browser en donde se visualiza la aplicación. Se conecta con el nodo de ElearningConsumidorREST por medio de una conexión HTTP.
 - ElearningConsumidorNET: este nodo representa a un consumidor del servicio, el cual está desarrollado en ASP.NET, con el fin de representar la interoperabilidad del sistema. Se comunica con el proveedor del servicio por medio de una conexión REST/HTTP.

4.4 Otros aspectos de la arquitectura

Es importante resaltar como el servicio desarrollado provee los links apropiados de modo que los clientes (en este caso la aplicación web) sólo necesitan una URL publicada para navegar a través del sistema, esto es, el cliente no requiere conocer detalles de como construir la URL de los servicios de negocio (como en SOAP) sólo requiere conocer la primera URL de donde inicia todo el proceso del protocolo de aplicación; las demás interacciones las arma el servidor y las devuelve como vínculos para que el cliente sólo realice la navegación y cambie el estado de la aplicación y del recurso.

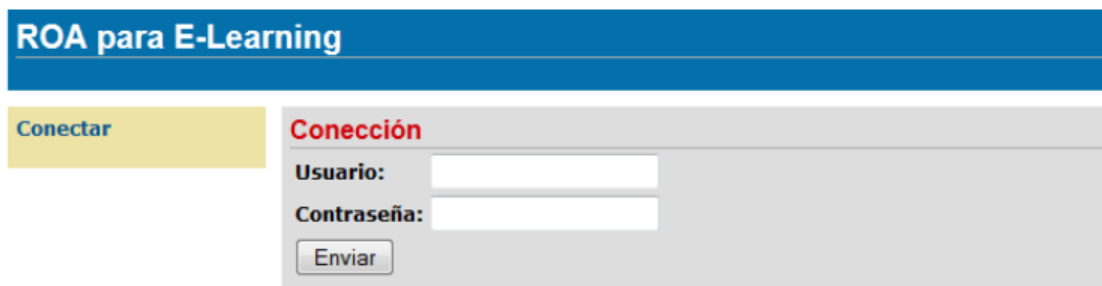
Como se presenta a lo largo del documento, una característica principal de la arquitectura orientada a recursos que hace parte de la solución es la actualización del estado de los mismos, por tal motivo esta soporta métodos CRUD (Create Read Update Delete) permitiendo modificar el estado de los recursos de otra manera. Para mayor información ver Anexo 5. Por otra parte en esta sección también se abordan las principales características a nivel de atributos de calidad de la solución propuesta, tales como escalabilidad e interoperabilidad. Para más profundización Anexo 5.

V – IMPLEMENTACIÓN DEL PROTOTIPO

Una vez definida las arquitecturas lógica y física, se procede a presentar la implementación final del prototipo. Como se mencionó en la sección anterior, el caso de estudio corresponde a una asignatura del Departamento de Ingeniería de Sistemas (Bases de Datos) y la tecnología utilizada fue Java E.E. 7.0. Adicionalmente se realizó la implementación de dos tipos de usuario, uno aprendiz y otro administrador.

5.1 Usuario Aprendiz

Inicialmente, el usuario debe ingresar al sistema digitando su nombre de usuario y contraseña, como se presenta en la siguiente imagen



The image shows a web interface for 'ROA para E-Learning'. At the top, there is a blue header with the text 'ROA para E-Learning'. Below the header, there is a yellow button labeled 'Conectar'. To the right of the button, there is a section titled 'Conección' in red. This section contains two input fields: 'Usuario:' and 'Contraseña:'. Below these fields is a button labeled 'Enviar'.

Ilustración 46 Ingreso al Sistema

Una vez el aprendiz ingresa al sistema, se le presentan el listado de cursos ofrecidos, como se ilustra en la siguiente imagen

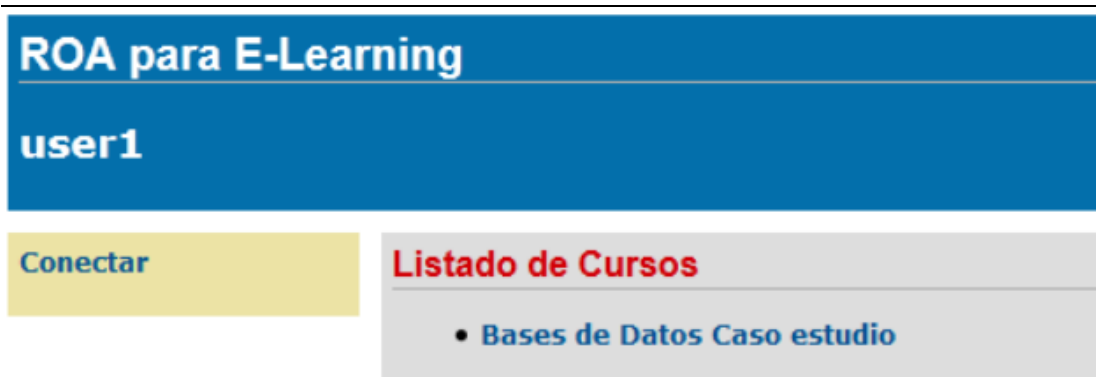


Ilustración 47 Listado de Cursos ofrecidos

El aprendiz selecciona el curso de interés, y a continuación se despliega el listado de unidades pertenecientes al curso seleccionado, como se presenta en la siguiente imagen

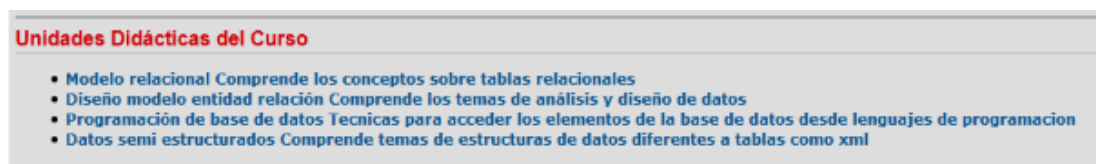


Ilustración 48 Unidades del curso seleccionado

Adicionalmente, se le presenta al usuario el camino que va siguiendo durante su proceso de autoaprendizaje. Tal como se ilustra en la siguiente imagen

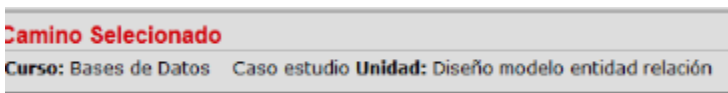


Ilustración 49 Camino tomado por el aprendiz

Una vez el aprendiz selecciona la unidad con la cual desea empezar, se procede a mostrarle la lista de objetivos de aprendizaje correspondientes, adicionalmente cabe aclarar que los objetivos alternos se presentan pero se encuentran deshabilitados hasta tanto el objetivo al que pertenecen cambie su estado a "Terminado". Como se ilustra en la siguiente imagen.

The screenshot shows the 'ROA para E-Learning' interface. At the top, there is a blue header with 'user1'. Below it, a yellow button labeled 'Conectar' is visible. The main content area has a light gray background and includes a link 'Volver al listado de Unidades'. Underneath, the 'Camino Seleccionado' is displayed as 'Curso: Bases de Datos Caso estudio Unidad: Diseño modelo entidad relación Comprende los temas de análisis y diseño de datos'. The 'Objetivos de Aprendizaje de la Unidad' section is highlighted in red and contains a list of objectives, all marked as '(NO INICIADO)':

- Entender diseño modelo entidad relacion EDMER (NO INICIADO)
 - Objetivos Alternos
 - Analizar diseño modelo entidad relacion ADMER (NO INICIADO)
 - Aplicar diseño modelo entidad relacion ADMER (NO INICIADO)
 - Entender modelo entidad relacion EMER (NO INICIADO)
 - Analizar modelo entidad relacion AMER (NO INICIADO)
 - Aplicar modelo entidad relacion AMER (NO INICIADO)
 - Entender diseño de bases de datos relacionales EDBDR (NO INICIADO)
 - Analizar diseño de bases de datos relacionales ADBDR (NO INICIADO)

Ilustración 50 Objetivos de aprendizaje de la unidad seleccionada

El aprendiz debe seleccionar el objetivo de aprendizaje por el cual desea comenzar, una vez lo hace se le presentan las actividades correspondientes. Las actividades alternas también son presentadas pero se encuentran deshabilitadas hasta tanto la actividad a la que pertenecen cambie su estado a “Terminado”. A continuación se presenta la imagen que ilustra lo mencionado anteriormente.

The screenshot shows the 'ROA para E-Learning' interface. At the top, there is a blue header with 'arning'. Below it, a yellow button labeled 'Volver al Listado de Objetivos' is visible. The main content area has a light gray background and includes a link 'Volver al Listado de Objetivos'. Underneath, the 'Camino Seleccionado' is displayed as 'Curso: Bases de Datos Caso estudio Unidad: Programación de base de datos Tecnicas para acceder los elementos de la base de datos desde lenguajes de programacion Objetivo: Aplicar programación de base de datos APBD'. The 'Actividades del Objetivo Seleccionado' section is highlighted in red and contains a list of activities, all marked as '(NO INICIADO)':

- Tutorial Slides (NO INICIADO)
 - Actividades Alternas
 - Proyecto grupal Enunciado (NO INICIADO)
- Proyecto grupal Enunciado (NO INICIADO)
 - Actividades Alternas

Ilustración 51 Actividades del objetivo seleccionado

Una vez el aprendiz selecciona la actividad a desarrollar, se le presenta el listado de OVAS que apoyan dicha actividad. El usuario puede comenzar por cualquiera de estos.

Por otra parte, como se mencionó con anterioridad un OVA representa un conjunto de Slides que tienen navegación entre ellos. Una vez el aprendiz selecciona el OVA que va a estudiar, comienza la navegación, como lo ilustra la siguiente figura.

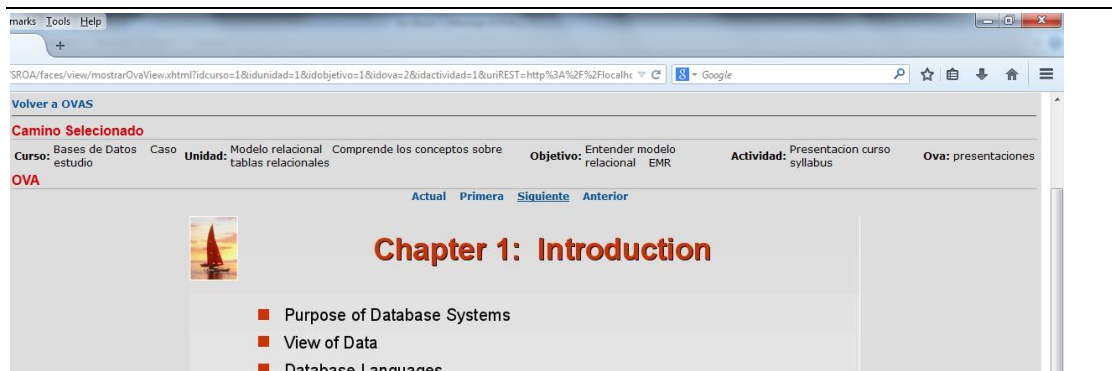


Ilustración 52 Navegación OVA seleccionado

Una vez el aprendiz finaliza todos los OVAS de la actividad, esta cambia su estado a “Terminado” y se habilitan las actividades alternas que posee, y una vez finalice todas las actividades, ha cumplido con el objetivo de aprendizaje razón por la cual este cambia su estado a “Terminado” y se habilitan los objetivos de aprendizaje alternos que posee.

5.2 Usuario administrador

Este usuario accede al sistema por el mismo medio que el usuario aprendiz (Ver Ilustración 46 Ingreso al sistema). Una vez ingresa los datos al sistema, se le habilita la opción administrar, como se ilustra en la siguiente imagen

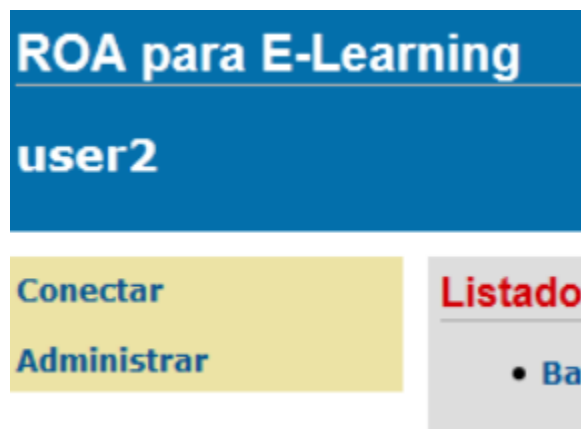


Ilustración 53 Opción administrar

Al ingresar a esta opción, el usuario tiene la posibilidad de modificar los OVAS ya existentes o crear nuevos. A continuación se despliega la lista de OVAS del sistema como se muestra en la siguiente imagen

The screenshot shows the 'ROA para E-Learning' interface for user 'user2'. On the left, there is a sidebar with 'Conectar' and 'Administrar' buttons. The main content area is titled 'Administración de Ovas del Sistema' and includes a link to 'Agregar Nuevo OVA'. Below this is a table titled 'Listado de Ovas del Sistema' with the following data:

Accion	Descripcion	Actividad	Slides
Administrar	presentaciones	syllabus	3
Administrar	laboratorioX	Enunciado	3
Administrar	nuevo ova	syllabus	10
Administrar	nuevo ova	syllabus	10
Administrar	nuevo ova 3	Enunciado	10
Administrar	nuevo ova 4	syllabus	5
Administrar	nuevo ova 5	Slides	5
Administrar	ova 11	syllabus	6
Administrar	slides 2	Enunciado	3
Administrar	Pruebas 2	Slides	3

Ilustración 54 OVAS del sistema

Al seleccionar el OVA que se desea administrar, el usuario puede editar la información correspondiente tal como se ilustra en la siguiente imagen

The screenshot shows the 'Administración de Ova' form. On the left, there is a yellow sidebar with 'Conectar' and 'Administrar' buttons. The main content area has a blue header with 'ROA para E-Learning' and 'user2'. Below the header, the form is titled 'Administración de Ova' and 'Crear/Editar'. It contains the following fields: 'Idova' with the value '5', 'Descripcion' with the value 'nuevo ova', 'Actividad' with a dropdown menu showing 'Presentacion curso syllabus', and 'Nslides' with the value '10'. At the bottom of the form, there are links for 'Eliminar[NO POSIBLE]', 'Modificar', and 'Volver'.

Ilustración 55 Editar OVA

Las modificaciones se pueden comprobar regresando al listado de OVAS del sistema y verificando que efectivamente la información se ha modificado.

Por otra parte el usuario puede crear otros OVAS ingresando a la opción “Agregar nuevo OVA” como se presenta en la siguiente imagen.

The screenshot shows the 'Administración de Ovas del Sistema' form. On the left, there is a yellow sidebar with 'Conectar' and 'Administrar' buttons. The main content area has a blue header with 'ROA para E-Learning' and 'user2'. Below the header, the form is titled 'Administración de Ovas del Sistema' and 'Agregar Nuevo OVA'.

Ilustración 56 Agregar nuevo OVA

Al ingresar por esta opción, debe diligenciar la información requerida y dar click en “agregar”

ROA para E-Learning

user2

Conectar

Administrar

Agregar Ova

Idova:

Descripcion:

ActividadIdactividad: ---

Nslides:

Agregar

Ilustración 57 Información nuevo OVA

ROA para E-Learning

user2

Conectar

Administrar

Agregar Ova

Idova: 15

Descripcion: TEST

ActividadIdactividad: Integridad Slides

Nslides: 9

Agregar

Ilustración 58 Información diligenciada

Para verificar que el OVA ha sido creado, el usuario debe retornar al listado de OVAS del sistema y comprobar que el OVA se encuentre allí tal como se ilustra a continuación

ROA para E-Learning

user2

Conectar

Administrar

Administración de Ovas del Sistema

Agregar Nuevo OVA

Listado de Ovas del Sistema

Accion	Descripcion	Actividad	Slides
Administrar	presentaciones	syllabus	3
Administrar	laboratorioX	Enunciado	3
Administrar	nuevo ova	syllabus	1
Administrar	nuevo ova	syllabus	10
Administrar	nuevo ova 3	Enunciado	10
Administrar	nuevo ova 4	syllabus	5
Administrar	nuevo ova 5	Slides	5
Administrar	ova 11	syllabus	6
Administrar	slides 2	Enunciado	3
Administrar	Pruebas 2	Slides	3
Administrar	TEST	Slides	9

Ilustración 59 OVA creado

VI - RESULTADOS Y REFLEXIÓN SOBRE LOS MISMOS

Este trabajo de grado cumplió con los objetivos propuestos. A continuación se enuncian los objetivos específicos y la forma en la que fueron cumplidos durante el desarrollo del trabajo.

1. Objetivo: Caracterizar el estilo arquitectural orientado a recursos en contextos de aprendizaje.

Para el cumplimiento de este objetivo se entrega:

- Un documento en donde se explica qué es una arquitectura orientada a recursos, componentes, relaciones y principios. (Anexo 1)
- Un documento con los objetos virtuales de aprendizaje utilizados en e-learning y su respectiva especificación. (Anexo 2)
- Un documento con la lista de objetos virtuales de aprendizaje seleccionados para ser representados en una arquitectura orientada a recursos y su respectiva representación. (Anexo3)

2. Objetivo: Diseñar un protocolo basado en recursos hipermediales para una aplicación e-learning.

Para el cumplimiento de este objetivo se entrega un documento donde especifica el proceso de aprendizaje seleccionado. (Anexo 4)

3. Objetivo: Diseñar una arquitectura dirigida por recursos que soporte el protocolo diseñado.

Para el cumplimiento de este objetivo se entrega un documento de arquitectura de software (S.A.D.), presentando arquitectura física y arquitectura lógica. (Anexo 5)

4. Objetivo: Validar la arquitectura mediante la implementación de un prototipo en el dominio de e-learning.

Para el cumplimiento de este objetivo se entregan los siguientes documentos:

- Documento con los objetos virtuales de aprendizaje generados y sus especificaciones. (Anexo 6)
- Documento con las especificaciones del prototipo a implementar. (Anexo 7)
- Adicionalmente se entrega el código fuente del prototipo implementado.

VII – CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

1. Conclusiones

- Se puede concluir que existe un nuevo paradigma para controlar el estado de una aplicación mediante el modelamiento de procesos usando principios de REST y ROA, específicamente el uso del HATEOAS.
- No se realizó una validación de la arquitectura con usuarios finales, ya que este proceso puede ser tomado como un trabajo de grado que complementa el presente, por este motivo se hizo una verificación de la solución por medio de un prototipo.
- Se ha probado la solución por medio de la implementación de un prototipo, utilizando la plataforma de JavaEE7, del cual fue posible concluir que el desarrollo en el cliente es más sencillo debido a que el servidor genera el vínculo requerido para contactar la lógica del protocolo de la aplicación (lógica de negocio del servicio).
- Durante el desarrollo de éste trabajo de grado, fue posible comprobar que el uso de servicios web tipo REST es más sencillo debido al uso de interfaz uniforme, que permite acceder a los recursos y consumir sus respectivas representaciones por medio de métodos HTTP, a diferencia de los servicios web tipo SOAP en donde uno de los limitantes es usa HTTP como medio de transporte y no como interfaz única.
- Es necesario enriquecer los servicios tipo REST con la implementación de protocolos de aplicación con el fin de agilizar el desarrollo de sistemas distribuidos, ya que se alinea TI (En la implementación del servicio) y el negocio (Del cual se puede abordar el protocolo de aplicación).
- Son pocas las aplicaciones que usan el HATEOAS por lo que se requiere desarrollar una metodología rigurosa de construcción de aplicaciones de este tipo.

2. Trabajos Futuros

A continuación se presentan posibles trabajos futuros relacionados con el presente trabajo de grado:

- Evolucionar los protocolos de aplicación hacia procesos de negocio usando BPM.
- Reflejar protocolos de aplicación de procesos de negocio en una arquitectura orientada a recursos enriqueciéndola con el uso de HATEOAS.
- Realizar una integración de la arquitecturas tipo ROA y SOA con el fin de complementarlas y lo lograr usar los beneficios que cada una acarrea.
- Diseñar una metodología para el desarrollo de aplicaciones que usen el paradigma de HATEOAS.
- Realizar la validación de la arquitectura solución, presentando el prototipo a usuarios finales.

VIII - REFERENCIAS Y BIBLIOGRAFÍA

- Andrea M, A. J. (2009). *Tecedu*.
- Aretio, L. G. (2006). *Características de la producción de materiales para la formación a distancia*. Madrid.
- Burke, B. (2013). *RESTful Java with JAX-RS2.0*. Sebastopol, California: O'Reilly.
- Cortés, J. C. (2009). Los Tres Escenarios de un Objeto de Aprendizaje. *Revista Iberoamericana de Educación*.
- Dodero, J. M., & Ghiglione, E. (2008). Rest Based Web Access to Learning Design Services. *IEEE Transactions on Learning Technologies*, 190-195.
- Fielding, R. T. (2000). *UNIVERSITY OF CALIFORNIA, IRVINE*. Obtenido de <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- Fielding, R. T. (2000). *UNIVERSITY OF CALIFORNIA, IRVINE*. Obtenido de http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- Jim Webber, S. P. (2010). *Rest in Practice*. Sebastopol: O'Reilly.
- Marset, R. N. (Julio de 2006). *Universidad Politécnica de Valencia*. Obtenido de <http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf>
- Michael zur Muehlana, J. V. (2005). Developing web services choreography standards - the case os REST vs SOAP. *Science Direct*, 9-29.
- Nations, F. a. (2011). *E-learning Methodologies A Guide for Design and Developing e-learning courses*. Roma: Fiat Panis.
- Papazoglou, M. P. (2008). SOAP: Simple Object Access Protocol. Tilburg, Holanda.
- Pautasso, C. (2007). SOAP vs REST Bringing the web back into web services. Zurich.
- Richard N. Taylor, N. M. (2010). Service-Oriented Architectures and Web Services. En N. M. Richard N. Taylor, *Software Architecture: Foundations, Theory and Practice*. Wiley.
- Ron Oliver, A. O. (2000). *Australian Society for Computers in Learning in Tertiary Education*. Obtenido de Australian Society for Computers in Learning in Tertiary Education: http://www.ascilite.org.au/conferences/coffs00/papers/ron_oliver2.pdf

ROY T. FIELDING, R. T. (2002). Principled Design of the Modern Web Architecture. *ACM Transactions on Internet Technology*, 115-150.

ruby, L. R. (2007). RESTful Web Services. En L. R. ruby, *RESTful Web Services*.

Savas Parastatidis, J. W. (2010). *researchgate*. Obtenido de www.researchgate.net/...The_role_of_hypermedia_in_distributed_system_development

Wu, C., & Chang, E. (24 de Mayo de 2007). aligning with the web: an atom-based architecture for web services discovery. Londres, Reino Unido.

Xiwei, X., Liming, Z., Yang, L., & Staples, M. (2010). Resource Oriented Architecture for Business Processes. *NICTA*, 4.

IX - ANEXOS

Anexo 1. Documento sobre arquitectura orientada a recursos

Anexo 2. Documento con objetos virtuales de aprendizaje en e-learning

Anexo 3. Documento con objetos virtuales de aprendizaje para una arquitectura orientada a recursos en e-learning

Anexo 4. Documento con proceso de aprendizaje seleccionado

Anexo 5. S.A.D (software Architecture document)

Anexo 6. Documento con objetos virtuales de aprendizaje creados y su especificación correspondiente

Anexo 7. Documentación del prototipo

Anexo 8. Creación de un curso e-learning (Caso de estudio)

Anexo 9. Glosario

- ROA: Resource Oriented Architecture

-
- REST: Representational State Transfer
 - SOA: Service Oriented Architecture
 - SOAP: Simple Object Access Protocol
 - HATEOAS: Hypermedia as the Engine of Application State

ANEXO 2

CARTA DE AUTORIZACIÓN DE LOS AUTORES (Licencia de uso)

Bogotá, D.C., Agosto 1 de 2014

Señores
Biblioteca Alfonso Borrero Cabal S.J.
Pontificia Universidad Javeriana
Cuidad

Los suscritos:

Angélica María Vergara Granados , con C.C. No 1032444177
_____, con C.C. No _____
_____, con C.C. No _____

En mi (nuestra) calidad de autor (es) exclusivo (s) de la obra titulada:
Arquitectura Orientada a Recursos para Aplicaciones Hipermediales en Contextos E-learning

(por favor señale con una "x" las opciones que apliquen)
Tesis doctoral Trabajo de grado Premio o distinción: Si No
cual: _____

presentado y aprobado en el año 2014 , por medio del presente escrito autorizo (autorizamos) a la Pontificia Universidad Javeriana para que, en desarrollo de la presente licencia de uso parcial, pueda ejercer sobre mi (nuestra) obra las atribuciones que se indican a continuación, teniendo en cuenta que en cualquier caso, la finalidad perseguida será facilitar, difundir y promover el aprendizaje, la enseñanza y la investigación.

En consecuencia, las atribuciones de usos temporales y parciales que por virtud de la presente licencia se autorizan a la Pontificia Universidad Javeriana, a los usuarios de la Biblioteca Alfonso Borrero Cabal S.J., así como a los usuarios de las redes, bases de datos y demás sitios web con los que la Universidad tenga perfeccionado un convenio, son:

AUTORIZO (AUTORIZAMOS)	SI	NO
1. La conservación de los ejemplares necesarios en la sala de tesis y trabajos de grado de la Biblioteca.	X	
2. La consulta física (sólo en las instalaciones de la Biblioteca)	X	
3. La consulta electrónica - on line (a través del catálogo Biblos y el Repositorio Institucional)	X	
4. La reproducción por cualquier formato conocido o por conocer	X	
5. La comunicación pública por cualquier procedimiento o medio físico o electrónico, así como su puesta a disposición en Internet	X	
6. La inclusión en bases de datos y en sitios web sean éstos onerosos o gratuitos, existiendo con ellos previo convenio perfeccionado con la Pontificia Universidad Javeriana para efectos de satisfacer los fines previstos. En este evento, tales sitios y sus usuarios tendrán las mismas facultades que las aquí concedidas con las mismas limitaciones y condiciones	X	

De acuerdo con la naturaleza del uso concedido, la presente licencia parcial se otorga a título gratuito por el máximo tiempo legal colombiano, con el propósito de que en dicho lapso mi (nuestra) obra sea explotada en las condiciones aquí estipuladas y para los fines indicados, respetando siempre la titularidad de los derechos patrimoniales y morales correspondientes, de

acuerdo con los usos honrados, de manera proporcional y justificada a la finalidad perseguida, sin ánimo de lucro ni de comercialización.

De manera complementaria, garantizo (garantizamos) en mi (nuestra) calidad de estudiante (s) y por ende autor (es) exclusivo (s), que la Tesis o Trabajo de Grado en cuestión, es producto de mi (nuestra) plena autoría, de mi (nuestro) esfuerzo personal intelectual, como consecuencia de mi (nuestra) creación original particular y, por tanto, soy (somos) el (los) único (s) titular (es) de la misma. Además, aseguro (aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos de la Tesis o Trabajo de Grado es de mí (nuestro) competencia exclusiva, eximiendo de toda responsabilidad a la Pontificia Universidad Javeriana por tales aspectos.

Sin perjuicio de los usos y atribuciones otorgadas en virtud de este documento, continuaré (continuaremos) conservando los correspondientes derechos patrimoniales sin modificación o restricción alguna, puesto que de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación de los derechos patrimoniales derivados del régimen del Derecho de Autor.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “*Los derechos morales sobre el trabajo son propiedad de los autores*”, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables. En consecuencia, la Pontificia Universidad Javeriana está en la obligación de RESPETARLOS Y HACERLOS RESPETAR, para lo cual tomará las medidas correspondientes para garantizar su observancia.

NOTA: Información Confidencial:

Esta Tesis o Trabajo de Grado contiene información privilegiada, estratégica, secreta, confidencial y demás similar, o hace parte de una investigación que se adelanta y cuyos resultados finales no se han publicado. Si No

En caso afirmativo expresamente indicaré (indicaremos), en carta adjunta, tal situación con el fin de que se mantenga la restricción de acceso.

NOMBRE COMPLETO	No. del documento de identidad	FIRMA

FACULTAD: Ingeniería

PROGRAMA ACADÉMICO: Ingeniería de Sistemas

ANEXO 3
BIBLIOTECA ALFONSO BORRERO CABAL, S.J.
DESCRIPCIÓN DE LA TESIS O DEL TRABAJO DE GRADO
FORMULARIO

TÍTULO COMPLETO DE LA TESIS DOCTORAL O TRABAJO DE GRADO						
Arquitectura Orientada a Recursos para Aplicaciones Hipermediales en Contextos E-learning						
SUBTÍTULO, SI LO TIENE						
AUTOR O AUTORES						
Apellidos Completos			Nombres Completos			
Vergara Granados			Angélica María			
DIRECTOR (ES) TESIS O DEL TRABAJO DE GRADO						
Apellidos Completos			Nombres Completos			
Carreño Vargas			Julio Ernesto			
FACULTAD						
Ingeniería						
PROGRAMA ACADÉMICO						
Tipo de programa (seleccione con "x")						
Pregrado	Especialización	Maestría	Doctorado			
X						
Nombre del programa académico						
Ingeniería de Sistemas						
Nombres y apellidos del director del programa académico						
Germán Alberto Chavarro Flórez						
TRABAJO PARA OPTAR AL TÍTULO DE:						
Ingeniero de Sistemas						
PREMIO O DISTINCIÓN (En caso de ser LAUREADAS o tener una mención especial):						
CIUDAD		AÑO DE PRESENTACIÓN DE LA TESIS O DEL TRABAJO DE GRADO			NÚMERO DE PÁGINAS	
Bogotá		2014			110	
TIPO DE ILUSTRACIONES (seleccione con "x")						
Dibujos	Pinturas	Tablas, gráficos y diagramas	Planos	Mapas	Fotografías	Partituras
		X				
SOFTWARE REQUERIDO O ESPECIALIZADO PARA LA LECTURA DEL DOCUMENTO						
<p>Nota: En caso de que el software (programa especializado requerido) no se encuentre licenciado por la Universidad a través de la Biblioteca (previa consulta al estudiante), el texto de la Tesis o Trabajo de Grado quedará solamente en formato PDF.</p>						

MATERIAL ACOMPAÑANTE					
TIPO	DURACIÓN (minutos)	CANTIDAD	FORMATO		
			CD	DVD	Otro ¿Cuál?
Vídeo					
Audio					
Multimedia					
Producción electrónica					
Otro Cuál?					
DESCRIPTORES O PALABRAS CLAVE EN ESPAÑOL E INGLÉS					
Son los términos que definen los temas que identifican el contenido. <i>(En caso de duda para designar estos descriptores, se recomienda consultar con la Sección de Desarrollo de Colecciones de la Biblioteca Alfonso Borrero Cabal S.J en el correo biblioteca@javeriana.edu.co, donde se les orientará).</i>					
ESPAÑOL			INGLÉS		
Hipermedia			Hypermedia		
Arquitectura Orientada a Recursos			Resource Oriented Architecture		
Transferencia de Estado Representativo			Representational State Transfer		
E-learning			E-learning		
RESUMEN DEL CONTENIDO EN ESPAÑOL E INGLÉS (Máximo 250 palabras - 1530 caracteres)					
<p>The main objective of this work is build a Resource Oriented Architecture supporting an application protocol (business process) using hipermedia as a principle between REST services and the Architecture mentioned at the begining, whitin e-learning context. Giv-ing and answer to what was mentioned before, it begins giving an explanation about the creation of a e-learning course with its principal elements, then is defined a learning process in e-learning context which will be the application protocol. Finally, it start de-fining an Resource Oriented Architecture, based on the learning process.</p> <p>El objetivo principal de este trabajo de grado es construir una arquitectura orientada a recursos soportando un protocolo de aplicación (proceso de negocio), haciendo uso de la hipermedia como principio fundamental entre los servicios REST y la arquitectura ya mencionada, todo enfocado al contexto de e-learning. Dando respuesta a lo menciona-do anteriormente, se comienza explicando la creación de un curso e-learning con sus principales elementos, posteriormente se define el proceso de aprendizaje adecuado para e-learning que será tomado como protocolo de aplicación. A partir de esto, se co-mienza a definir la arquitectura orientada a recursos, teniendo en cuenta el proceso de aprendizaje.</p>					