



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

Desarrollo de módulos administrativos de la Red Universitaria
de Aprendizaje(RUA)

REPORTE DE TRABAJO PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A:

Jonathan Leonardo Silva Blasio

TUTOR:

Dr. Gustavo De la Cruz Martínez

2019





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Hoja de datos del jurado

1. Datos del alumno
Silva
Blasio
Jonathan Leonardo
56 03 86 34
Universidad Nacional Autónoma de México
Facultad de Ciencias
Ciencias de la Computación
306041381
2. Datos del tutor
Dr.
Gustavo
De la Cruz
Martínez
3. Datos del sinodal 1
Dra.
Selene Marisol
Martínez
Ramírez
4. Datos del sinodal 2
Dra.
Hanna
Jadwiga
Oktaba
5. Datos del sinodal 3
M. en C.
Darío Emanuel
Vázquez
Ceballos
6. Datos del sinodal 4
M. en I.
Gerardo
Avilés
Rosas

Agradecimientos

Este trabajo y cada meta que cumpla es para la mejor persona que he conocido y tengo el privilegio de decirle mamá. Todos mis logros son gracias a todos los sacrificios que ella ha hecho por mí, espero que la vida le recompense todos esos sacrificios con más que el orgullo que siente por mí, a pesar de toda la distancia que exista entre los dos, cada paso que dé será gracias a ella.

Gracias mamá por brindarme todo tu apoyo y ahora es tiempo de que los sacrificios que hagas sean para mejor tu vida, por mi parte jamás dejaré de agradecerte por la persona que me hiciste.

Sin duda este trabajo no lo hubiera podido realizar sin todas las enseñanzas de los profesores que tuve a lo largo de mi trayectoria escolar, por lo cual siempre me llena de orgullo ser egresado de esta bella facultad de ciencias.

A pesar del tiempo que tarde en realizar este trabajo, mis amigos nunca dudaron de que lo lograría, por eso agradezco a mis compañeros con los que compartí clases sobre todo a Alan y Maya; y los nuevos amigos que encontré después en mi carrera profesional, los cuales siempre me animaron a terminar este ciclo de mi vida; de ellos a quien más agradezco es a mi hermano Michael.

Gracias a todo el equipo de trabajo de la DCV de DGTIC, que tuvieron la paciencia para desarrollar mis habilidades. Fue un placer trabajar a su lado. Sobre todo a Karla Fonseca que siempre tenía la respuesta correcta.

Índice general

Introducción	1
1. Preliminares	3
1.1. Antecedentes de la RUA	3
1.1.1. Motivación	4
1.1.2. Estructura de RUA	5
1.2. Ingeniería de software	8
1.3. Metodología del equipo de RUA	12
1.3.1. Metodología de trabajo	13
1.3.2. Versiones de RUA IES	15
1.4. Tecnologías usadas	18
1.4.1. PHP	18
1.4.2. CakePHP	19
1.4.3. PostgreSQL	20
1.4.4. Búsqueda natural de texto (Full-Text-Search) PostgreSQL	21
1.4.5. Consultas recursivas	22
2. Diseño	26
2.1. Especificación de Requerimientos RUA IES	26
2.1.1. Requerimientos módulo de administración	26
2.1.2. Requerimientos portal RUA IES	30
2.1.3. Requerimientos no funcionales RUA IES	31
2.2. Diagrama general de casos de uso	32
2.3. Funcionalidades de los módulos: listado recursos educativos y seguimiento de evaluaciones	33
2.3.1. Funcionalidades del módulo: listado de recursos educativos	33
2.3.2. Funcionalidades del módulo: seguimiento de evaluaciones	34
2.4. Reglas de asociación de elementos	35

2.5.	Arquitectura de RUA IES	37
2.5.1.	Base de datos	37
2.5.2.	Búsqueda en el programa de estudios de recursos educativos	40
2.6.	Estructura RUA IES	42
2.6.1.	Componentes de CakePHP	43
2.6.2.	Estructura RUA IES	44
2.7.	Vista estática RUA IES	45
2.7.1.	Diagrama de clases listado de recursos educativos	45
2.7.2.	Diagrama de clases seguimiento de evaluaciones de recursos educativos	47
2.8.	Vista dinámica RUA IES	48
2.8.1.	Diagrama de secuencia del módulo: listado de recursos educativos	48
2.8.2.	Diagrama de secuencia del módulo: seguimiento de evaluaciones	51
3.	Implementación	56
3.1.	Detalle de la implementación del módulo: Seguimiento de evaluaciones	56
3.1.1.	Modelo	56
3.1.2.	Controlador	57
3.1.3.	Vistas	58
3.2.	Implementación de la funcionalidad: Ingreso seguimiento de evaluaciones	58
3.3.	Implementación de la funcionalidad: Búsqueda de evaluaciones de recursos educativos	61
3.4.	Búsqueda de evaluaciones de recursos educativos	64
3.4.1.	Implementación función get_relaciones_descendencia	65
4.	Conclusiones	70
4.1.	Resumen general	70
4.2.	Experiencia en la metodología de trabajo	71
4.3.	Comentarios sobre el uso de tecnologías	71
4.4.	Relación con mi carrera	72
4.5.	Trabajo a futuro	72
A.	Casos de uso	I
A.1.	Caso de uso Administración de evaluaciones de recursos educativos	I
A.2.	Caso de uso Reporte listado de recursos educativos	VII
	Referencias	XII

Introducción

La Red Universitaria de Aprendizaje (RUA) UNAM es una plataforma que permite a la comunidad universitaria y público en general acceder a un acervo de contenidos educativos en línea: públicos, abiertos y gratuitos; dichos recursos están relacionados explícitamente con los componentes de los planes de estudios, tanto de bachillerato como de licenciatura de la UNAM. La RUA UNAM le permite al alumno encontrar recursos educativos relacionados con temas específicos, dentro de un plan de estudios, le indica los conocimientos previos que debe tener para abordar el tema, así como los recursos que le pueden ayudar a reforzar los conocimientos del tema; autoevaluarse y brinda acceso a otros recursos educativos que le permitan conocer más acerca del tema. Para ampliar las características de la RUA UNAM a distintas Instituciones de Educación Superior (IES) surge el proyecto de RUA IES. Permitiendo adaptarse a los planes de estudios que cada institución requiera.

En este documento se detalla el desarrollo de dos apartados del módulo administrativo de la RUA IES, brindando un panorama general a las diversas funcionalidades del sistema y su implementación. Se presentará el diseño e implementación de los módulos: Seguimiento de evaluación de recursos educativos y Listado de recursos educativos; del módulo administrativo.

Participación

Mi participación en la RUA fue como desarrollador de software, durante tres periodos del desarrollo de la RUA, el primero abarcando del 1° de Agosto al 11 de Diciembre del 2015; el segundo del 18 de Enero al 30 de Junio del 2016 y finalmente el último del 1° de Agosto al 30 de Noviembre del 2016. Mis principales actividades en este proyecto fueron:

- Estudiar y analizar la documentación técnica para la comprensión y desarrollo de los requerimientos, estos documentos incluyen: especificación de requerimientos, casos de uso, diagramas de clase, diagramas de secuencia y diagramas de entidad relación.

- Desarrollo de módulos y componentes para cumplir con los requerimientos del proyecto.
- Pruebas funcionales en la etapa de desarrollo para garantizar el correcto funcionamiento de los componentes y módulos implementados.
- Pruebas de integración de los componentes desarrollados con módulos desarrollados por los demás miembros del equipo de desarrolladores.
- Solución de defectos, incidencias y nuevos requerimientos reportados por el equipo interno de pruebas a través de la herramienta Mantis.
- Creación de manuales de usuario e instalación de la RUA IES para permitir su configuración en los servidores de las distintas IES.

En este reporte se describe el trabajo realizado dentro del proyecto, en las tres fases en las que participé, con la descripción de los módulos antes mencionados.

Organización

La organización de este reporte es la siguiente:

En el primer capítulo se describirán los antecedentes de la RUA y la motivación del nuevo desarrollo; se dará una breve introducción a las etapas de un proceso de desarrollo de software así como la metodología utilizada para el desarrollo del proyecto. Adicionalmente se describirán las principales tecnologías utilizada en la implementación del mismo. Todo esto con el objetivo de brindar una visión general del proyecto de RUA.

En el segundo capítulo se mostrarán los distintos módulos que componen el sistema; se describirá la estructura general de la RUA, para entender el contexto del diseño de dos módulos; de los cuales se detallarán sus componentes y la interacción entre estos; basándose en los conceptos de diseño descritos en el primer capítulo.

En el tercer capítulo se revisará la implementación de una parte específica de funcionalidad del sistema. Con esto también buscando profundizar en la descripción de las tecnologías utilizadas.

Finalmente en el cuarto capítulo, se presentan las conclusiones de este reporte.

1. Preliminares

En este reporte se hablará del desarrollo de la Red Universitaria de Aprendizaje (RUA), comenzando por describir el sitio original del proyecto, mencionando la motivación del nuevo desarrollo y las ventajas de este; En este documento se describirán dos módulos en los que colaboré, para entender el funcionamiento de la propuesta.

En este reporte denotaremos como RUA Azul al sitio web que precedió a la RUA IES, el cual fue sustituido en el año 2017; y RUA UNAM a la instancia de RUA IES cuya configuración está adaptada para ser usada por la UNAM.

1.1. Antecedentes de la RUA

La Red Universitaria de Aprendizaje (RUA) Azul surge como un sitio web desarrollado para brindar acceso a recursos educativos¹ (Objetos de Aprendizaje), disponibles en línea, generados por los profesores de la UNAM u otros autores; para fomentar la creación de estos recursos y así apoyar el estudio independiente del alumno. La RUA Azul permitía desarrollar, consolidar y poner a disposición de la comunidad universitaria y sociedad en general, recursos educativos en línea con relación a los planes de estudios² de la UNAM, estos recursos se pueden encontrar en distintos formatos, como: audios, videos, ejercicios, textos, interactivos, presentaciones, etc.; para que el usuario elija el que mejor se adapte a sus necesidades.

La RUA Azul mostraba los planes de estudios de bachillerato (CCH y ENP) y planes de estudios de algunas carreras de la UNAM, dichos planes de estudios se dividieron en cuatro niveles: ciclos (años, semestres), materias, unidades y temas; a partir de esta división se asociaban los aprendizajes esperados, aprendizajes previos, sub aprendizajes y recursos

¹“**Recurso educativo:** Son materiales digitales concebidos para que alumnos y maestros se acerquen a los contenidos de los programas de estudio, para promover la interacción y el desarrollo de las habilidades digitales, el aprendizaje continuo y para que los estudiantes logren su autonomía” [3, p.63].

²“**Plan de estudios:** Es la referencia sintética, esquematizada y estructurada de las asignaturas u otro tipo de unidad de aprendizaje” [3, p. 63].

educativos de cada tema.

La RUA Azul brindaba recursos educativos organizados por los distintos planes de estudios de la UNAM, para facilitar a los alumnos encontrar recursos educativos, de acuerdo con el tema del plan de estudios del cual deseen aprender.

Un sitio como RUA fomenta la creación de recursos educativos en línea, hace accesible los recursos educativos generados por los profesores, permite clasificar el contenido digital existente de la universidad, guía al alumno en un aprendizaje autónomo, permitiendo evaluar su conocimiento y habilidades y, promueve el estudio independiente, ya que se ajusta al tiempo del alumno.

La RUA UNAM hasta el 2015 contaba con un total de 9920 recursos educativos, distribuidos en los distintos planes de estudios de la siguiente manera[3]:

- Escuela Nacional Preparatoria (ENP): 3574
- Colegio de Ciencias y Humanidades (CCH): 1679
- Licenciatura: 3597
- Temas transversales: 1070

1.1.1. Motivación

Debido a la diversidad entre los planes de estudios de cada Institución de Educación Superior (IES) que existe en México la RUA Azul (al solo contemplar cuatro niveles dentro del programa de estudios³), no puede ser adaptada para brindar los beneficios antes mencionados a las distintas IES que existen en el país.

De esta manera surge el proyecto RUA IES, que dará la oportunidad de configurar los programas de estudios de cada IES con los niveles y nomenclatura que cada institución requiera,; para ello se propone representar la información de los planes de estudios a través de una estructura de árboles, para satisfacer las distintas configuraciones que pueda requerir cada IES.

La RUA IES deberá ser extensible y parametrizable para su uso y aprovechamiento en otras instituciones y entidades académicas del país, bajo la filosofía de cooperación y acceso abierto a los recursos educativos conforme a los planes de estudios de cada institución para beneficio de la sociedad en general.

Se podrá contar con recursos educativos organizados y asociados a los planes y programas de estudios para apoyar cursos presenciales, mixtos y en línea. Además de contener material confiable, libre y gratuito para el público en general, también puede contener material privado para la comunidad de la institución.

³“**Programa de estudios:** Es la descripción detallada y sintetizada de los contenidos de las asignaturas, ordenadas por secuencias o por áreas, es decir, brinda información particular de cada asignatura que conforma un plan de estudios”[3, p 62].

La RUA IES es un proyecto desarrollado por la Dirección General de Cómputo y Tecnologías de Información y Comunicación (DGTIC), Dirección de Colaboración y Vinculación (DCV), Subdirección de Sistemas Integrados.

1.1.2. Estructura de RUA

El proyecto RUA IES se divide en dos módulos, el módulo administrativo y el módulo de consulta, que a continuación se describen:

1. El módulo de administración se encarga del registro, edición, eliminación y búsqueda de los distintos elementos que componen la RUA IES. En este módulo se configuran los recursos educativos, programas de estudios y aprendizajes esperados que podrán ser visualizados por la comunidad escolar de cada IES, así como por el público general. Este módulo solo brinda acceso a los administradores del sistema que podrán ser añadidos en el mismo; con distintos perfiles para la administración de la RUA IES.
2. El módulo de consulta (portal) RUA IES brinda acceso a los elementos capturados en el módulo de administración; además permite la participación de los estudiantes para mejorar y dar sugerencias a estos contenidos. Este módulo es accesible por el público general y solo se limita el acceso a los recursos educativos que sean privados para los alumnos de la institución.

El módulo de consulta lo conforman dos sub-módulos principales, que son: la navegación a través del programa de estudios para obtener los distintos recursos educativos y aprendizajes esperados; y el buscador principal de recursos educativos que permite buscar todos los recursos educativos que cumplan con los criterios ingresados por el alumno.

El portal considera un diseño web responsivo para ser accesible desde distintos navegadores de internet así como de dispositivos móviles, para la comodidad del alumno.

Los módulos representan dentro de RUA una funcionalidad necesaria para brindar los beneficios del sitio mencionados con anterioridad. Los principales módulos y componentes en los cuales participé dentro de la RUA IES son descritos a continuación:

- **Administración de aprendizajes esperados**

Permite administrar los aprendizajes esperados mediante la captura, la edición, la eliminación y búsqueda de la información.

Los aprendizajes esperados pueden estar asociados a cualquier nodo del árbol del programa de plan de estudios, así como tener varias relaciones de los nodos con un aprendizaje.

- **Personalización de la identidad de la RUA IES**

Permite modificar el nombre, las siglas, el tema del portal y del módulo de administración y el logotipo que se presenta en la interfaz gráfica de la RUA; para personalizarlo

de acuerdo con la identidad de la IES que haga uso del sistema.

Los datos capturados en esta sección permiten realizar la catalogación de los recursos educativos, programas de estudios y aprendizajes esperados de acuerdo a la IES que se encargó de crear o catalogar estos elementos.

- **Reporte de listado de recursos educativos**

Permite consultar un listado con la información de los recursos educativos capturados, además de la búsqueda de estos recursos a través de múltiples criterios de búsqueda. Los campos mostrados en este listado pueden ser seleccionados por el usuario de acuerdo a la información que desee obtener de estos para comprobar la información de cada recurso.

- **Reporte de estadísticas de recursos educativos**

Permite generar el reporte que presenta la cantidad de recursos educativos que se encuentran asociados en cualquier nivel del programa de estudios. Estos recursos se agrupan para saber el número de recursos que se encuentran en un nivel específico del plan de estudios; ya sea de manera inmediata o explorando el árbol. Este reporte además permite filtrar los recursos educativos por múltiples criterios de búsqueda referentes a los campos de los recursos educativos.

- **Reporte de estatus de URLs de recursos educativos**

Permite consultar el estatus de las URL de los recursos educativos de tipo URL que se encuentran catalogados en la RUA IES.

Estos recursos pueden ser filtrados a través de tres categorías que son:

1. Planes de estudios, filtra los recursos educativos de tipo URL que se encuentran relacionados con el plan de estudios seleccionado.
2. Categoría del recurso, filtra los recursos educativos a través de la categoría en la cual se encuentran catalogados, estas categorías son: autoevaluación, recurso para aprender y recurso para profesor.
3. Estado de publicación del recurso, que puede ser publicado y no publicado.

- **Seguimiento de evaluación de recursos educativos**

Permite a los administradores dar seguimiento a las evaluaciones de los recursos educativos realizados por los profesores registrados para esta actividad en la RUA IES; el administrador puede ver el detalle de la evaluación del recurso educativo y editar las propiedades del recurso en caso de que se presente una observación que permita mejorar la catalogación del recurso educativo o su vigencia.

Además permite la búsqueda de evaluaciones por distintos campos, correspondientes de los recursos educativos que fueron evaluados o a los profesores que evaluaron los recursos educativos. Este reporte puede ser exportado a Microsoft Excel.

A continuación describiremos otras componentes desarrollados, como parte de mis actividades, que están presentes en distintos módulos de la RUA IES:

- **Manual LOM-RUA**

Permite a los catalogadores de recursos educativos conocer los metadatos que se encargan de describir las características más relevantes de los recursos educativos. Dichas características se encuentran agrupadas en ocho categorías: General, Ciclo de vida, Técnica, Uso educativo, Derechos, Clasificación, Control y Capturas. Estos metadatos están adaptados a partir de la adaptación del estándar LOM (IEEE Std 1484.12.1 for Learning Object Metadata).

- **Detalle del recurso educativo**

Permite visualizar los datos capturados hasta el momento de un recurso educativo, agrupado en las ocho distintas categorías establecidas en el Manual LOM de RUA. Este componente es utilizado para tener una visión general del recurso educativo que se asociará a un elemento (plan de estudios, elemento del programa de estudios y recursos educativos).

- **CRON de verificación de estatus URLs de los recursos educativos**

Permite la verificación periódica del estatus de la URL de los recursos educativos de tipo URL registrados en la RUA IES. Esto se realiza a través del envío de múltiples peticiones HTTP de las URL registradas en los recursos, desde el servidor en el que se encuentra instalada la instancia de la RUA IES.

Este proceso permite mantener actualizado el estado de publicación de los recursos, ya que no se mostrarán en el portal aquellos recursos cuyo estatus de URL no sea accesible.

- **Rutas del programa de estudio**

Se implementa la visualización de las rutas a partir del plan de estudios, hacia cualquier elemento del programa de estudios (plan de estudios, elemento del programa de estudios o recurso educativo).

Además de la búsqueda en los planes de estudios en los cuales se encuentra asociados un elemento (nodo, recurso educativo y aprendizaje esperado).

Esta implementación contempla obtener cualquier elemento catalogado dentro del plan de estudios a través de una búsqueda por anchura (*Breadth First Search* BFS) y búsqueda por profundidad (*Depth First Search* DFS); la búsqueda a profundidad es utilizada únicamente en el módulo del portal.

Este componente es utilizado en distintos módulos de la RUA IES tales como búsquedas en los distintos módulos administrativos, reportes, búsqueda principal de recursos educativos y navegación del programa de estudios.

En este reporte se describirán detalladamente solo los módulos: *Reporte de listado de recursos educativos* y *Seguimiento de la evaluación de recursos educativos*; los cuales representan

el aporte más significativo de mis actividades al sistema, y desde los cuales podemos entender el funcionamiento general de la RUA y sus distintos componentes, por la similitud que existe entre estos.

1.2. Ingeniería de software

La Ingeniería de software es la disciplina que estudia los aspectos de la creación del software; desde su inicio, con la especificación (alcance y límites); hasta el mantenimiento y evolución del mismo; dentro de esta disciplina se desarrollan distintas herramientas para el desarrollo de software como son: modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de los procesos. La meta de esta disciplina es el desarrollo costeable del software en términos económicos y temporales[1]. En el contexto de la ingeniería de software “el software es el conjunto de programas, documentos asociados y configuración de datos, para la correcta operación de un sistema”[1, p. 5]. Algunos de los atributos considerados para crear un software de calidad son: funcionalidad, rendimiento requerido por el usuario, mantenibilidad, confiabilidad, usabilidad, eficiencia y aceptabilidad.

Para lograr el objetivo de esta disciplina existen un conjunto de actividades las cuales nos conducen a un producto de software; a este conjunto lo llamamos proceso de software, estas actividades pueden dividirse en cuatro fases fundamentales, presentes en la mayoría de los procesos de software:

1. Especificación

Es el proceso en el cual se identifica la funcionalidad que cumplirá el sistema así como las restricciones en su funcionamiento. A las funcionalidades y restricciones del sistema les llamamos requerimientos; estos son obtenidos a partir de las solicitudes del usuario de los servicios que debe proveer el sistema (*requerimientos de usuario*). En esta etapa deben de ser descritos y documentados las funciones, servicios y restricciones que el sistema debe tener (*requerimientos del sistema*). Los requerimientos del sistema se dividen en: **requerimientos funcionales**, los cuales son la especificación de cómo el sistema debe comportarse ante entradas determinadas; y **requerimientos no funcionales**, los cuales se refieren a las restricciones de la funcionalidad brindada por el sistema, estas pueden darse por cuestiones de: tiempo, proceso de desarrollo o estándares definidos.

En esta fase se debe estudiar la viabilidad del sistema, comprobarse que las necesidades del usuario puedan cumplirse con las tecnologías actuales y se pueda desarrollar dentro de un presupuesto acordado. La información brindada por el usuario sobre la funcionalidad que debe de cumplir se traduce en un documento llamado especificación de requerimientos; una vez obtenido ese documento es necesario validar la consistencia de los requerimientos y queden comprendidas las funcionalidades requeridas por el

usuario.

2. Diseño y desarrollo

Para realizar este proceso se debe diseñar y desarrollar el software; estas tareas están fuertemente ligadas. El diseño nos ayuda a generar modelos del software con distintos niveles de abstracción que nos darán una base para iniciar su desarrollo.

En la fase de diseño se identifican y describen, con base en los requerimientos descritos en la etapa anterior, los componentes necesarios para cumplir con la especificación del sistema. En esta fase también se describe la interacción que debe de existir entre los distintos componentes que comprenden el sistema. Para lograr este objetivo se realiza un análisis del sistema desde distintos niveles de abstracción[1]. En esta sección describiremos tres niveles, los cuales nos ayudaran a describir más adelante los distintos componentes de RUA.

Arquitectura del sistema

En esta etapa del diseño se identifican los componentes que conformarán el sistema para cumplir con los requerimientos funcionales y no funcionales, así como las relaciones que existen entre ellos. El modelo generado de la arquitectura del sistema no debe de cambiar considerablemente con el cambio o adición de requerimientos.

Vista estática

Con este modelo identificamos las distintas clases necesarias para cumplir con los requerimientos del sistema; cada clase es descrita en términos de sus atributos y métodos; además de las relaciones que existen entre ellas. Esto lo podemos representar por medio de un diagrama de clases, donde opcionalmente las clases son agrupadas en paquetes.

Vista dinámica

Este modelo nos sirve para mostrar la interacción entre los objetos a través de sus métodos, para cumplir con los requerimientos del sistema. Este modelo lo podemos representar a través de **diagramas de secuencia** los cuales se componen por:

Actor: El cual es un ente externo al sistema que realiza una interacción en el sistema para iniciar el flujo de la funcionalidad.

Objetos: Son los representantes de la clase, los cuales se representan con un rectángulo y tienen una línea de tiempo.

Mensajes: Información que se va enviando de un objeto a otro, en el caso de los objetos, los mensajes pueden ser enviados a través de la invocación de métodos que disparen la acción de un objeto distinto[2].

Derivado de los distintos modelos, se codifica el software en un lenguaje de progra-

mación particular, que cumpla con los requerimientos del sistema descritos en la fase de la especificación.

3. Verificación y validación

Es la fase en la cual se demuestra que el software producido en la fase de diseño y desarrollo cumple con los requerimientos especificados por el usuario. Esta fase también es utilizada para encontrar comportamiento incorrecto en el software, o un comportamiento que no cumple con los requerimientos establecidos.

Para lograr este objetivo se realizan distintos tipos de pruebas en distintas etapas del proceso de software, las principales son:

Pruebas en desarrollo: Estas pruebas las realiza el equipo de desarrollo durante la fase de diseño y desarrollo y pueden ser:

- Pruebas unitarias
- Pruebas de integración
- Pruebas del sistema

Pruebas de lanzamiento: Estas pruebas son realizadas por un equipo distinto al de desarrollo, este equipo prueba una versión completa del sistema y su principal objetivo es encontrar errores de programación y verificar que el sistema cumpla con los requerimientos establecidos.

Pruebas de usuario: Estas pruebas las realizan los usuarios en el ambiente, donde el software será utilizado; el principal objetivo es validar que el software es adecuado para que los usuarios lo utilicen.

Las pruebas realizadas principalmente en las pruebas de lanzamiento y pruebas de usuario las llamamos pruebas de caja negra o pruebas funcionales, en las cuales se verifica una funcionalidad del sistema sin conocer el detalle de su implementación.

4. Evolución

Esta fase comienza con la instalación del software, para que los usuarios hagan uso de él. Con el sistema en operación pueden surgir errores que no fueron detectados en la etapa de pruebas o nuevos requerimientos identificados por los usuarios para darle mayor utilidad al software.

Derivado de los cambios en los requerimientos que surgen durante la operación del software; se genera un ciclo, donde se reinicia el proceso de desarrollo de software para atender los nuevos requerimientos encontrados[1].

Estas cuatro etapas pueden ser implementadas de distintas maneras, a continuación describiremos dos modelos del proceso de software.

Modelo en cascada

En este modelo se consideran las cuatro fases fundamentales del proceso de software de manera independiente, por lo que una etapa no se inicia hasta que se concluya con la etapa previa. En este modelo las fases de especificación, diseño y desarrollo, validación y evolución corresponden a: la especificación de requerimientos, diseño de software, implementación, pruebas y mantenimiento respectivamente. En la figura 1.1 se presenta un diagrama del modelo en cascada.

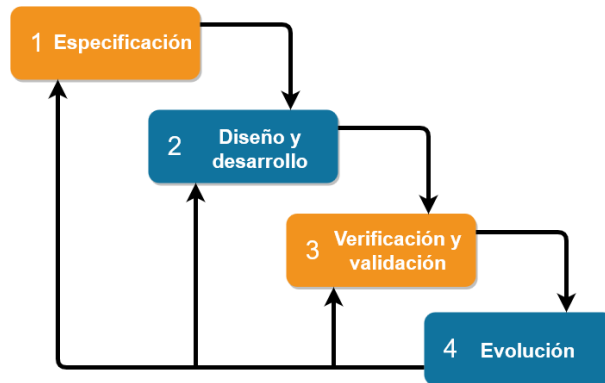


Figura 1.1: Modelo en cascada [1]

Desarrollo incremental

En este modelo, en cada etapa se añade una parte de la funcionalidad del sistema, priorizando cada funcionalidad, de acuerdo con las necesidades del usuario, con lo cual se generan varias versiones de él. En cada nueva entrega se agrega una nueva funcionalidad. En este modelo las fases de especificación, diseño y desarrollo, y validación se realizan de forma conjunta. Existen cuatro principales ventajas comparado con el modelo en cascada:

1. El costo en el desarrollo de los cambios en los requerimientos dados por el usuario es menor, debido a que el análisis y documentación del sistema se reduce a la versión de la entrega realizada.
2. Se obtiene en menor tiempo la retroalimentación del usuario al realizar pequeñas entregas.
3. Se tiene un desarrollo y entrega de software útil, sin que se termine por completo con todos los requerimientos del sistema, lo que permite a los usuarios, su aprovechamiento y uso en un menor tiempo que en un modelo de cascada.

4. Se realizan más pruebas en componentes críticos del sistema; esto si tienen una prioridad alta, ya que son desarrollados al inicio del proceso de software.

Dentro de este modelo existen dos principales problemas, los cuales se acentúan en sistemas grandes, complejos o de larga vida, o donde se tienen distintos equipos desarrollando distintas partes del sistema.

1. Si los sistemas son desarrollados rápidamente, no es costeable generar la documentación para cada versión, lo cual produce opacidad en el proceso del software.
2. La incorporación de nuevos requerimientos, o cambios en estos, se vuelven más costosos en etapas avanzadas del proceso. Esto puede producir el decaimiento en la calidad del software si no se invierte tiempo en el ajuste del código para realizar los cambios.

En la figura 1.2 se presenta un diagrama del modelo incremental.



Figura 1.2: Desarrollo incremental

1.3. Metodología del equipo de RUA

En esta sección describiremos el proceso de software utilizado para el desarrollo de la RUA IES, los equipos que participaron y la dinámica entre ellos; así como de las distintas fases de desarrollo en las cuales participé y sus principales diferencias.

Mi participación en el proyecto de RUA IES fue dentro del equipo de desarrollo durante las tres primeras versiones, en cada fase del proyecto se contó con tres equipos que trabajaban de manera conjunta en el proceso de software: el equipo de análisis, el equipo de desarrollo y el equipo de aseguramiento de la calidad.

- **Equipo de análisis:** Se encargaba, principalmente, de obtener los requerimientos del sistema, generando la documentación y diseño del mismo. El resultado de su trabajo era compartido con los otros equipos para tener la misma visión del proyecto.

Adicionalmente cada que se terminaba la implementación de un caso de uso, el equipo de análisis realizaba las pruebas del sistema.

- **Equipo de desarrollo:** Se encargaba de implementar los requerimientos generados por el equipo de análisis, el desarrollo se basaba principalmente en la implementación de los casos de uso generados, así como la integración de los distintos módulos del sistema.
- **Equipo de aseguramiento de la calidad:** Se encargaban de hacer las pruebas de lanzamiento, sus pruebas aseguraban que la documentación generada y el sistema desarrollado coincidieran.

En la figura 1.3 se presenta un diagrama del equipo de RUA.

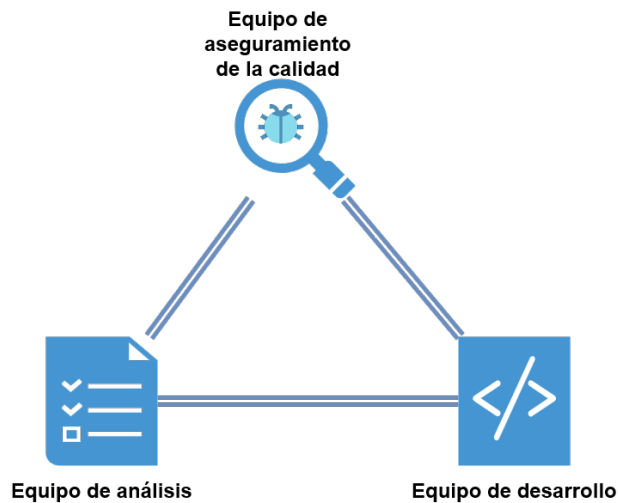


Figura 1.3: Equipo RUA

1.3.1. Metodología de trabajo

A continuación se describirá la interacción entre los equipos de trabajo. Lo primero era seleccionar un conjunto de módulos de RUA que serían implementados en el ciclo de desarrollo, de estos módulos el equipo de análisis generaba la especificación de requerimientos, casos de uso, prototipos de la interfaz y el diagrama entidad relación asociados a estos módulos.

Una vez generados, estos documentos eran compartidos con el equipo de desarrollo para tener una visión general de lo que se iba a implementar. Cada caso de uso era asignado a un desarrollador para su implementación. Una vez que eran aprobada la documentación

de cada módulo y se iniciaba el desarrollo, el equipo de análisis comenzaba con la creación de la documentación para el siguiente ciclo de desarrollo.

Durante el desarrollo de los módulos se tenía una constante comunicación entre los equipos de análisis y desarrollo para la actualización de la documentación, en caso de ser necesarias; o una actualización en el código debido a un cambio en la documentación; esta comunicación se daba principalmente entre los líderes de cada equipo los cuales compartían la información con los desarrolladores y analistas.

Cada desarrollador debía de integrar el módulo implementado dentro de un repositorio, validando que no tuviera errores y no generará conflictos con los demás módulos cargados anteriormente en el repositorio.

Una vez terminado el desarrollo de un módulo por parte de un desarrollador, un integrante del equipo de análisis realizaba pruebas del sistema para corroborar que el módulo no tuviera errores y coincidiera con la documentación generada. Si existieran errores o la implementación no coincide con la documentación, estas incidencias son reportadas al desarrollador para su corrección de manera inmediata. En la figura 1.4 presentamos un ciclo de desarrollo.

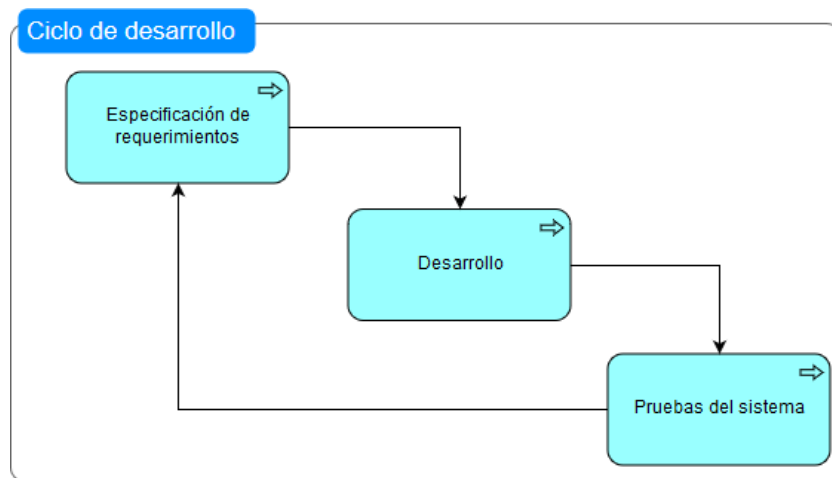


Figura 1.4: Ciclo de desarrollo RUA

Estos ciclos de desarrollo se repetían hasta tener una versión estable del sistema, con un conjunto de módulos con los cuales el usuario podía interactuar con el sistema, una vez obtenida esta versión, en promedio en 4 ciclos de desarrollo, se generaba una versión para el equipo de aseguramiento de la calidad.

Por otro lado, el equipo de aseguramiento de la calidad revisaba la documentación del sistema generada por el equipo de análisis y generaba el plan de pruebas para el sistema. Los errores del sistema e incongruencias entre la documentación y el software encontradas por el equipo de aseguramiento de la calidad eran reportadas a través de una herramienta

llamada **Matis Bug Tracker**, como incidencias principalmente se documentaba los pasos para reproducir la incidencia, una imagen como evidencia de que ocurrió y bajo qué circunstancias se presenta. Las incidencias podían tener los siguientes estatus: Creada, Abierta, Resuelta y Cerrada.

Las incidencias primero eran asignadas al líder del equipo de aseguramiento de la calidad, el cual verificaba que la incidencia se presentará y cambiaba su estatus a *Abierta*. Una vez *Abierta* la incidencia se pasaba al líder del equipo de desarrollo, el cual asignaba el reporte al desarrollador que le correspondía el módulo en el que se presentó la incidencia.

Al finalizar el ciclo de pruebas se procedía a corregir las incidencias encontradas. Cada desarrollador cambiaba el estatus de sus incidencias asignadas a *Resuelta* cuando eran corregidas. Una vez corregidas todas las incidencias se actualizaba la versión de pruebas y se pedía al equipo de aseguramiento de la calidad verificar que las incidencias hubieran sido corregidas correctamente; Si la incidencia ya no se presentaba se cambiaba su estatus a *Cerrada* en caso contrario se cambiaba el estatus a *Abierta*. Los ciclos de pruebas se repitieron en promedio tres veces para generar una versión de RUA IES. En la figura 1.5 presentamos un ciclo de pruebas del sistema.

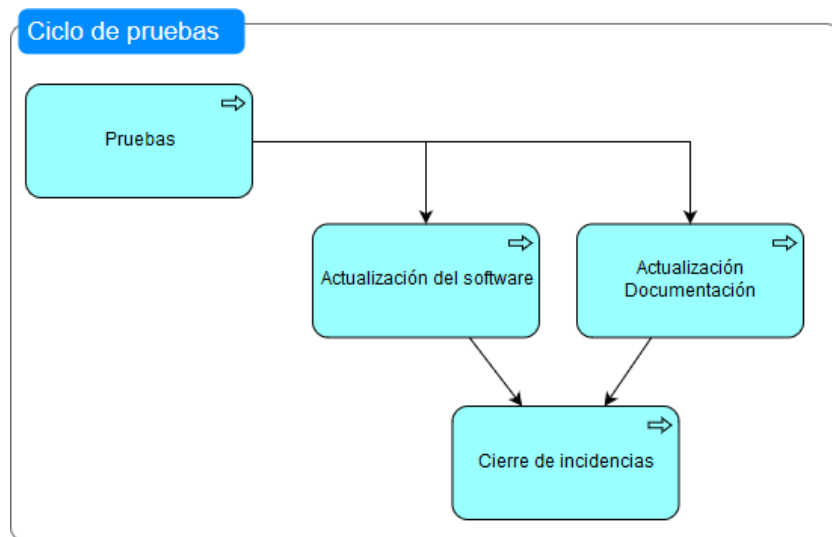


Figura 1.5: Ciclo de pruebas RUA

1.3.2. Versiones de RUA IES

RUA IES se desarrolló en tres fases principales, en cada una de estas fases se generó una versión estable del sistema, en cada fase se fueron incrementando las funcionalidades de la RUA IES para cumplir con las especificaciones de la RUA Azul. A continuación se detallarán los principales objetivos de cada versión de la RUA IES.

■ Versión 1

El principal objetivo de la primera versión de RUA IES era brindar flexibilidad en la creación de los programas de estudios, para extender los beneficios de la RUA Azul a otras IES. Se incluyó la búsqueda de los recursos educativos y aprendizajes esperados en cualquier elemento del programa de estudios, así como hacer accesibles los distintos recursos educativos catalogados.

Los principales módulos de la RUA IES que fueron implementados dentro de esta versión son: la administración de planes y programas de estudios, recursos educativos, aprendizajes esperados, usuarios, reportes de recursos educativos, personalización de la RUA IES, búsqueda de recursos educativos y relaciones entre los elementos de la RUA IES (planes de estudios, recursos educativos y aprendizajes esperados). En la figura 1.6 se presenta el portal de la RUA IES en su primera versión.

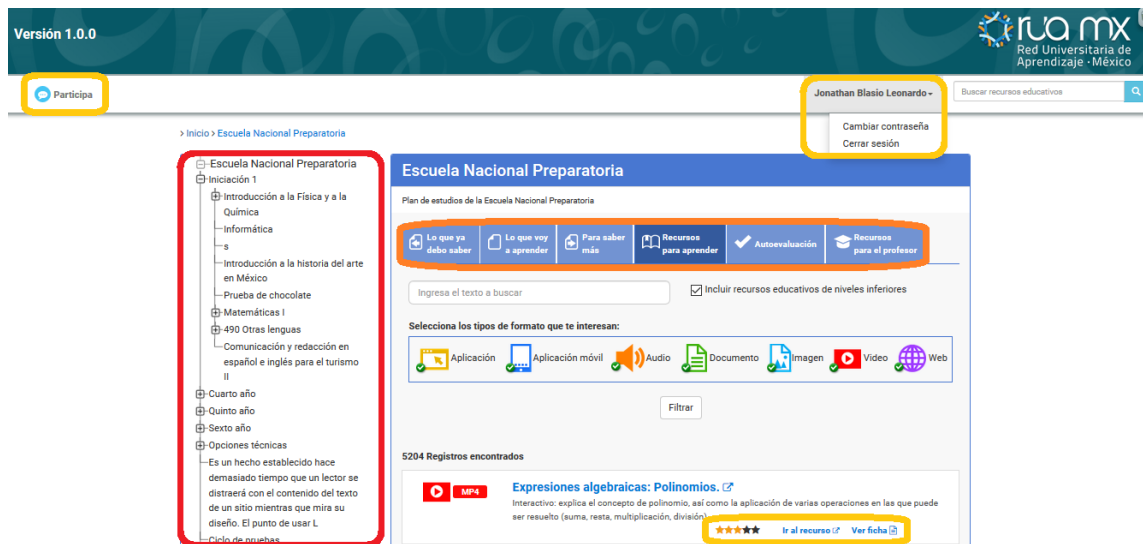


Figura 1.6: Portal RUA IES versión 1

■ Versión 2

En esta versión se reconstruyó la interfaz de usuario en la búsqueda de recursos educativos, además se homogeneizó la implementación de los distintos módulos de la RUA IES para tener un comportamiento similar. Se cambiaron y restringieron las reglas de asociaciones entre los elementos de la RUA IES y se realizó un proyecto alterno llamado RUA MX para concentrar todos los elementos de las distintas instancias de RUA IES.

■ Versión 3

En esta versión se agregaron funcionalidades de la RUA IES, presentes en la RUA

Azul, que no habían sido consideradas en las versiones iniciales tales como: evaluación de recursos educativos por profesores de las IES, recomendaciones de recursos educativos y una autenticación específicamente para la RUA UNAM por medio del servicio web definido en `servicios.unam.mx`.

Adicionalmente, en esta versión se trabajó la migración de los datos actuales de la RUA IES, tales como: recursos educativos y programas de estudios de la UNAM, para realizar el cambio de la RUA Azul a la RUA UNAM. Los cambios y funcionalidades nuevas de esta versión fueron enfocados para subir a un ambiente productivo correspondiente a la RUA IES para la UNAM. En la figura 1.7 se presenta el portal de la RUA IES en su versión 3.

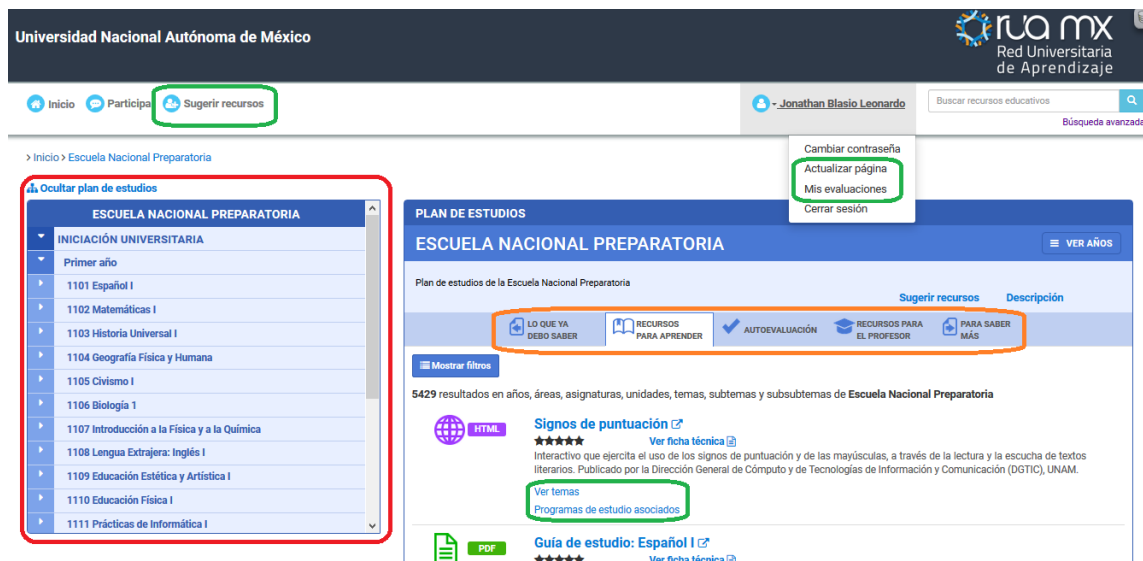


Figura 1.7: Portal RUA IES versión 3

En la Tabla 1.1 se pueden observar los módulos que han sido implementados en cada versión.

Módulo	Versión 1	Versión 2	Versión 3
Autenticación.	✓	✓	✓
Administrar usuarios.	✓	✓	✓
Administrar usuarios de comunidad escolar.		✓	✓
Administrar perfiles y privilegios de usuario.	✓	✓	✓
Personalizar identidad de la RUA.	✓	✓	✓
Administrar aprendizajes esperados.	✓		
Administrar planes de estudios.	✓	✓	✓
Administrar programas de estudios.	✓	✓	✓

Módulo	Versión 1	Versión 2	Versión 3
Administrar asociaciones de elementos.	✓	✓	✓
Administrar recursos educativos.	✓	✓	✓
Buscar recursos educativos.	✓	✓	✓
Navegar por plan de estudios.	✓	✓	✓
Consultar ficha técnica.	✓	✓	✓
Consultar listado de recursos educativos.	✓	✓	✓
Consultar estadísticas de recursos educativos.	✓	✓	✓
Consultar estatus de URLs.	✓	✓	✓
Registrar comentarios y sugerencias.	✓	✓	✓
Configurar estructura del plan de estudios.	✓	✓	✓
Importar archivo con datos de usuario.		✓	✓
Consultar información del sistema.			✓
Asignar profesores a elementos del programa de estudios.			✓
Administrar página del profesor.		✓	✓
Evaluar recursos educativos.			✓
Sugerir recursos educativos.			✓

Tabla 1.1: Módulos implementados de la RUA IES

1.4. Tecnologías usadas

En esta sección describiremos las principales tecnologías utilizadas: lenguajes de programación, sistemas manejadores de base de datos, frameworks y bibliotecas; para el desarrollo de la RUA IES.

1.4.1. PHP

PHP es un lenguaje de programación interpretado, de código abierto, que puede ser incrustado en HTML, fue creado en 1994 por Rasmus Lerdorf[7], utilizado por 79 % de los sitios web en 2018[14]. Es un lenguaje de propósito general aunque es utilizado principalmente como un lenguaje del lado del servidor para realizar páginas web dinámicas. El código de PHP es ejecutado en el servidor web generando un documento de HTML, el cual es enviado como respuesta al cliente; por esta razón el cliente recibirá el resultado de la ejecución de los scripts y sin conocer el código que genera dicho resultado[6].

1.4.2. CakePHP

CakePHP es un framework de desarrollo rápido libre y abierto para PHP, que sigue los principios del patrón de diseño Modelo-Vista-Controlador (MVC)[8]. Contiene herramientas comunes para el desarrollo de aplicaciones web. La tarea principal de CakePHP es permitir trabajar de una manera rápida y estructurada sin perder flexibilidad. CakePHP provee de una estructura organizacional que cubre, nombres de clases, archivos, nombres de tablas de bases de datos, y otras convenciones. Esto permite omitir configuración innecesaria y tener una estructura uniforme en la aplicación que hace el desarrollo más simple. A continuación tenemos la filosofía que sugiere el framework de la implementación del patrón de diseño MVC.

- **La capa del Modelo**

Representa la parte de la aplicación que implementa la lógica del negocio. Es responsable de obtener datos y convertirlos en conceptos significativos en la aplicación.

- **La capa de la Vista**

Genera una representación de los datos modelados. Siendo separada de los objetos del modelo, es responsable de usar la información que tiene disponible para producir cualquier interfaz de presentación que la aplicación pueda necesitar.

- **La capa del Controlador**

Maneja las peticiones del usuario. Es responsable de generar una respuesta con ayuda de las capas del Modelo y la Vista al usuario[8].

A continuación describiremos como CakePHP maneja las peticiones para la utilización del patrón de diseño MVC[9]:

1. El cliente realiza una petición a nuestro sistema
2. El Despachador (Dispatcher) de cake transforma los datos de la URL solicitada en los parámetros: controlador, acción y argumentos.
3. El Ruteador (Router) mapea el controlador y acción extraídos de la URL para su ejecución.
4. El Controlador puede usar Modelos para extraer información de la base de datos
5. El Modelo regresa los datos solicitados por el controlador.
6. El controlador puede utilizar Componentes (Components) para realizar tareas como: manipular los datos de sesión, hacer uso de los datos de autorización, mandar correos electrónicos; por mencionar algunas.

7. Una vez que la información es extraída de la base de datos y procesada por el Controlador esta es enviada a la vista. La lógica de la vista se ejecuta, esta puede incluir el uso de Elementos (Elements) y/o Ayudantes (Helpers). Las vistas dentro de CakePHP regularmente se colocan sobre una plantilla (Layout).
8. El resultado de la vista es enviado al usuario como respuesta.

En la figura 1.8 se presenta un diagrama con lo descrito anteriormente.

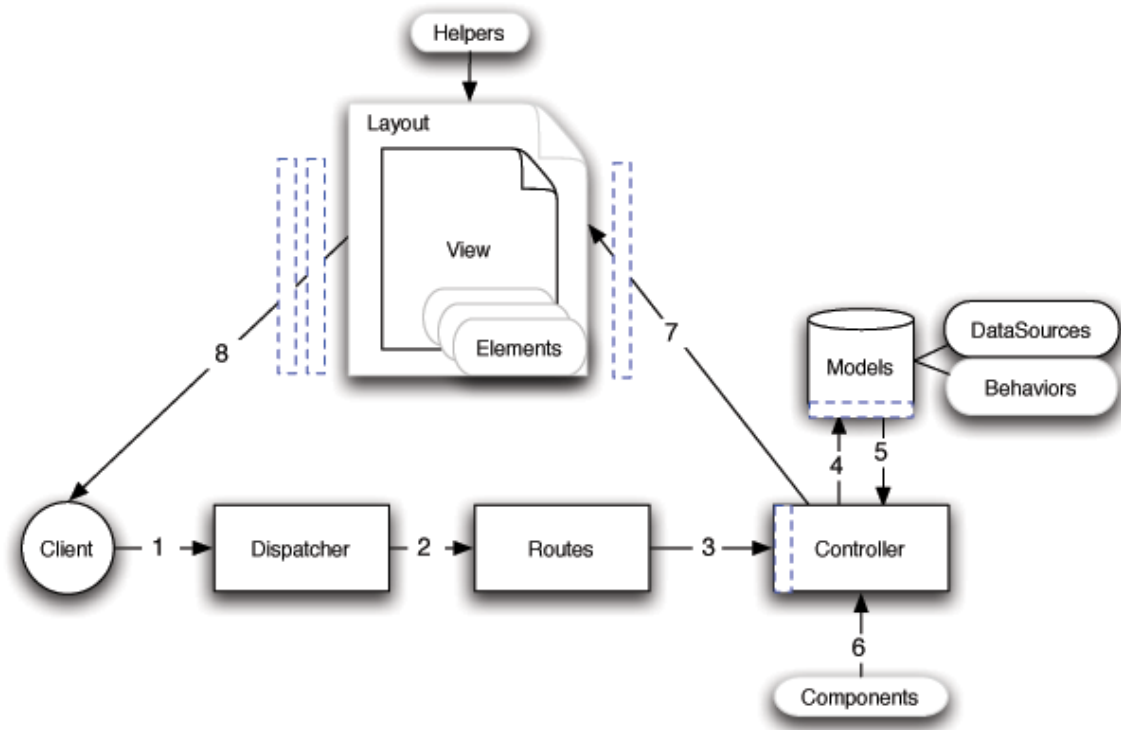


Figura 1.8: Petición CakePhp [9, img typical-cake-request.png]

1.4.3. PostgreSQL

PostgreSQL es un sistema manejador de base de datos relacional de código abierto desarrollado en 1986 por la Universidad de California en Berkley. PostgreSQL utiliza el lenguaje SQL para manipular los datos, sus transacciones cumple con las propiedades **ACID** (Atomicidad, Consistencia, Aislamiento y Durabilidad) aunado con la comunidad activa detrás del software PostgreSQL ha ganado una buena reputación por: su arquitectura, confiabilidad, integridad de los datos y extensibilidad[10].

Dentro de RUA utilizamos dos características brindadas por PostgreSQL: *búsqueda natural de texto* y *consultas recursivas*; las cuales describiremos a continuación.

1.4.4. Búsqueda natural de texto (Full-Text-Search) PostgreSQL

La búsqueda natural de texto (Full-Text-Search) es una característica de PostgreSQL que brinda la capacidad de buscar documentos en lenguaje natural que satisfacen una consulta y ordenarlos por relevancia. Un **documento** es la unidad sobre la cual se realizará la búsqueda natural de texto, este puede ser cualquier campo de una relación o un valor construido dinámicamente con la información de otras relaciones.

“La búsqueda más simple considera una consulta como un conjunto de palabras, y la similitud, como la frecuencia en la que se encuentran las palabras de la consulta dentro del documento” [11].

Las búsquedas textuales son implementadas en PostgreSQL a través del uso de los operadores como \sim , \sim^* , **LIKE**, e **ILIKE**; las desventajas de este tipo de búsqueda sobre la búsqueda natural de texto son:

- No hay soporte lingüístico. Las expresiones regulares no son suficientes ya que no pueden manejar palabras derivadas; por ejemplo: biología y biólogo.
- No brindan un orden (clasificación) de los resultados, que las hace menos eficientes cuando se encuentran miles de resultados derivados de una consulta.
- Tienden a ser lentas al no poder pre-procesar la información almacenada para agilizar la búsqueda.

En la búsqueda natural de texto los documentos son preprocesados para generar **índices**, para brindar una búsqueda más rápida. Las operaciones realizadas para crear los índices incluyen:

- Dividir los documentos en tokens⁴. Posteriormente se identifica el tipo de cada token; estos tipos pueden ser: números, palabras, palabras complejas, direcciones de correo electrónico, entre otros. Cada clase se debe de procesar de manera distinta. Las clases de tokens dependen de la aplicación.
- Convertir tokens en lexemas. Un lexema es un token que ha sido normalizado para que distintas formas de la misma palabra sean similares. La normalización incluye: sustituir letras mayúsculas por minúsculas, remover sufijos y prefijos, des-acentuación de vocales y la eliminación de palabras de paro, estas son palabras muy frecuentes que no son útiles para la búsqueda. En PostgreSQL se usan diccionarios para cumplir con esta tarea.

⁴**Token:** Conjunto de caracteres que tiene un significado per se.

- “Ordenar documentos preprocesados, para optimizar las búsquedas. Cada documento es representado como un arreglo normalizado de lexemas. Junto con los lexemas se guarda la posición en el documento de las palabras, para realizar una clasificación por proximidad, de esta manera un documento que contenga una región más densa de palabras de la consulta tiene una mayor similitud que uno con palabras dispersas” [11].

Los **diccionarios** permiten tener un control de cómo los tokens son normalizados. Con ellos puedes:

- Definir palabras de paro que no deben de ser procesadas.
- Asociar sinónimos a una misma palabra.
- Asociar frases a una simple palabra.
- Asociar distintas variaciones de una palabra a una forma simplificada.

Las bases del FTS en PostgreSQL son: los **diccionarios**, estos son provistos por PostgreSQL y pueden ser modificados con distintas operaciones; y los tipos de datos **tsvector** y **tsquery**, con los cuales podemos realizar las búsquedas con en el operador **@@**, que regresa verdadero si un **tsvector**(documento) coincide con un **tsquery**(consulta), sin importar en qué orden se encuentren estos tipos de datos[11]. Un **tsvector** es un tipo de datos que corresponde a una lista de lexemas junto con la posición de estos, asociados a un documento. Un **tsquery** representa un conjunto de lexemas los cuales serán buscados, este tipo de dato permite combinar los lexemas con los operadores[12]:

Símbolo	Operador
&	AND
	OR
!	NOT

PostgreSQL brinda las funciones **to_tsvector** y **to_tsquery** para transformar cadenas en los tipos de datos **tsvector** y **tsquery** respectivamente, para de esta manera realizar las búsquedas sobre los distintos campos de las relaciones.

1.4.5. Consultas recursivas

PostgreSQL implementa las consultas recursivas con el uso de **WITH RECURSIVE**, esta sentencia nos permite referirnos a los resultados de una misma consulta.

La forma de crear una consulta recursiva con el uso de **WITH RECURSIVE** es iniciar con una consulta no recursiva, seguido de **UNION** (o **UNION ALL**) y posteriormente un término recursivo, donde puede referirse a los datos de la salida de la consulta anterior. A continuación presentamos un ejemplo de una consulta recursiva:

```

1 WITH RECURSIVE t(n) AS (
2     VALUES (1)
3 UNION ALL
4     SELECT n + 1 FROM t WHERE n < 100
5 )
6 SELECT sum(n) FROM t
7 ;

```

Consulta 1.1: Ejemplo consulta recursiva

A continuación presentaremos cómo se evalúan las consultas recursivas tal como se describe en la documentación oficial de PostgreSQL[11]:

1. Se evalúa el término no recursivo. En el caso de **UNION** (pero no para **UNION ALL**), descarta renglones repetidos. Se incluyen todos los renglones restantes en el resultado de la consulta recursiva, y los coloca en una “*tabla temporal de trabajo*”.
2. Hasta que la tabla temporal de trabajo no esté vacía, repetir los siguientes pasos:
 - a) Evaluar el término recursivo, sustituyendo el contenido de la “*tabla temporal de trabajo*” en la referencia recursiva. Para **UNION** (pero no para **UNION ALL**), descarta renglones duplicados y renglones que duplican cualquier dato previamente obtenido. Se incluyen todos los renglones restantes en el resultado de la consulta recursiva, y los coloca en una “*tabla temporal intermedia*”.
 - b) Se reemplaza el contenido de la “*tabla temporal de trabajo*” con el contenido de la “*tabla temporal intermedia*”, después se vacía la “*tabla temporal intermedia*”.

Este tipo de consultas nos permiten realizar búsquedas sobre los datos que se encuentran en una estructura jerárquica como un árbol o gráfica.

Cuando usamos consultas recursivas debemos asegurar que la consulta recursiva termine, esto quiere decir que se debe de asegurar que dejemos de obtener datos; ó en caso de trabajar con **UNION** que los datos obtenidos en la consulta recursiva sean todos repetidos, para de esta forma garantizar que la “*tabla temporal de trabajo*” se encuentre vacía. Al trabajar con una estructura de gráficas estas pueden contener ciclos los cuales al añadir información adicional a los resultados de la consulta, esta haga que se cicle.

```

1 WITH RECURSIVE grafica_recorrido(id, nodo, data, profundidad) AS (
2     SELECT g.id, g.nodo, g.data, 1
3     FROM grafica g
4 UNION ALL
5     SELECT g.id, g.nodo, g.data, gr.profundidad + 1
6     FROM
7         grafica g

```

```

8      JOIN grafica_recorrido gr ON g.id = gr.nodo
9    )
10   SELECT * FROM grafica_recorrido
11  ;

```

Consulta 1.2: Consulta recursiva con ciclo indeterminado

La consulta 1.2 se ciclará en caso que la gráfica contenga un ciclo, para evitar que esto pase podemos agregar datos adicionales a nuestra consulta recursiva para poder recordar el camino que hemos recorrido y de esta manera evitar la expansión de un nodo que ya hemos visitado. Para lograr esto debemos almacenar los nodos que ya hemos recorrido(**camino**) y una variable booleana que nos indique si el nodo ya fue visitado(**visitado**).

```

1  WITH RECURSIVE grafica_recorrido
2    (id, nodo, data, profundidad, camino, visitado) AS (
3    SELECT g.id, g.nodo, g.data, 1,
4           ARRAY[g.id],
5           false
6    FROM grafica g
7  UNION
8    SELECT
9       g.id, g.nodo, g.data, gr.profundidad + 1,
10      gr.camino || g.id,
11      g.id = ANY(camino)
12  FROM
13     graph g,
14     JOIN grafica_recorrido gr ON g.id = gr.nodo
15  WHERE
16     NOT visitado
17  )
18  SELECT * FROM grafica_recorrido
19  ;

```

Consulta 1.3: Consulta recursiva sin ciclos indeterminados

En la consulta 1.3 podemos observar dos cosas en la parte *no recursiva*: la columna **camino**, en la cual agregamos en un arreglo el nodo que estamos expandiendo; y la columna **ciclo**, que la inicializamos en falso, ya que ese nodo no ha sido visitado con anterioridad. Después en la parte *recursiva* de la consulta vemos que en la columna **camino** vamos añadiendo los nuevos nodos que estamos visitando, agregando el **id** del nodo visitado al arreglo; y en la columna **visitado** comprobamos si el nuevo **id** del nodo que vamos a expandir ya ha sido visitado con anterioridad, comprobando si ya se encuentra en el **camino**. Adicionalmente comprobamos que el nodo no se encuentre **visitado**, para no incluirlo en caso de que ya haya sido expandido con anterioridad. Podemos observar que los valores de la columna camino representan los nodos recorridos para llegar a un punto[13].

Resumen

En este capítulo describimos los objetivos de la RUA azul y sus limitaciones; por lo cual se decidió crear una nueva versión de esta manera fuera aprovechada por distintas IES. También logramos describir los dos módulos principales de la RUA IES, y la aportación que realice en este proyecto.

Describimos los aspectos principales en los procesos de software para posteriormente, describir el proceso de software del proyecto. Adicionalmente describimos las herramientas principales utilizadas en el desarrollo de la RIA IES que nos ayudarán a entender la implementación; la cual detallaremos en el capítulo 3. El principal objetivo del capítulo es dar una base para entender el desarrollo de la RUA IES y poder entender el diseño que se hará del proyecto en el siguiente capítulo así como la descripción de la implementación en el último capítulo. Una de las características más importantes de la RUA IES que debemos de tener en mente a lo largo de este reporte es la búsqueda de estudios de recursos educativos por el programa.

2. Diseño

En este capítulo describiremos el diseño de la RUA IES, desde una perspectiva general y finalmente enfocándonos en la funcionalidad de los módulos: *Listado de Recursos Educativos* y *Seguimiento de Evaluación de Recursos Educativos*.

2.1. Especificación de Requerimientos RUA IES

En esta sección listaremos los distintos módulos que componen la RUA IES, marcando con ‘*’ los módulos desarrollados por mí, a continuación se presentan los requerimientos de estos módulos. Toda la información presentada en este trabajo sobre el funcionamiento del sistema está contenida y detallada en el documento de la especificación de requerimientos de la RUA IES [3], el cual fue creado por el equipo de análisis de la RUA. Para los fines de este trabajo, en esta sección se presenta una breve descripción del objetivo de cada uno de los módulos que conforman el sistema.

Como se mencionó con anterioridad, la RUA IES cuenta con dos módulos principales el Portal y la Administración, por lo cual dividiremos los requerimientos funcionales de la RUA IES en dos secciones correspondientes a cada módulo.

2.1.1. Requerimientos módulo de administración

Administración de usuarios y perfiles

1. **Autenticación:** Este módulo se encarga de restringir el acceso al módulo de administración de RUA a través de un correo electrónico y una contraseña, solo a los usuarios que se encuentran registrados y activos en el sistema; adicionalmente permite la recuperación de la contraseña en caso de que el usuario lo requiera.
2. **Administración de usuarios:** Este módulo permite el registro, edición, búsqueda y eliminación de usuarios administradores. Los usuarios registrados son los que podrán

tener acceso al sistema por medio del módulo de autenticación.

3. **Administración de perfiles y privilegios:** Este módulo permite la creación, edición, búsqueda y eliminación de perfiles. Los perfiles se componen de un conjunto de privilegios y un nombre. Los permisos pueden ser asignados a un usuario, lo que le permitirá acceder a los módulos del sistema de acuerdo a los privilegios que tenga el perfil del usuario.
4. **Administración de usuarios de la comunidad escolar:** Este módulo permite el registro, edición, búsqueda y eliminación de usuarios de la comunidad escolar. Los usuarios de la comunidad escolar son aquellos que pueden iniciar sesión en el portal y pueden ser de tipo alumno o profesor.
5. **Importar usuarios de comunidad escolar:** Este módulo permite el registro de usuarios de la comunidad escolar por medio de la carga de un archivo.
6. **Administración de profesores:** Este módulo permite asignar profesores a asignaturas del programa de estudios, esto permitirá listar los profesores que imparten una materia específica.

Configuración de RUA

1. **Personalización de la identidad de la RUA*:** Este módulo permite cambiar el logotipo que se presentará en todas las páginas de la RUA IES, además del nombre de la institución a la que pertenece la RUA. A continuación se presentan los requerimientos de este módulo.

Permitirá editar la información: Nombre de IES, correo institucional, siglas, sitio web, logotipo, tema y texturas.

Al guardar la información se presentará en el portal y módulo de administración: el logotipo y el nombre de la IES.

El acceso a este módulo deberá ser restringido sólo a los usuarios con permiso de personalizar identidad de la RUA.

2. **Configurar la estructura del plan de estudios:** Este módulo permite el registro, edición y ordenamiento de los niveles que formarán los programas de estudios según lo requiera la IES.

Administración de Planes y Programas de estudios

1. **Administración de planes de estudios:** Este módulo permite la creación, edición, búsqueda y eliminación de planes de estudios. Las acciones adicionales a las antes mencionadas que se pueden realizar en este módulo son: acceso al programa de estudios del plan de estudios y administración de asociaciones.

2. **Administración de programas de estudios:** Este módulo permite la creación, edición, búsqueda, ordenamiento y eliminación de elementos que conforman el programa de estudios. Para acceder a este módulo debemos seleccionar el plan de estudios del cuál queremos editar su programa de estudios.

Administración Recursos Educativos

1. **Administración de recursos educativos:** Este módulo permite la creación, edición, búsqueda y eliminación de recursos educativos. Adicionalmente permite al usuario guardar la información capturada sin necesidad de que esta información esté completa, para continuar posteriormente con su captura. En este módulo también se encuentra la administración de asociaciones.
2. **Seguimiento de evaluación de recursos educativos*:** Este módulo permite administrar la evaluaciones realizadas por los profesores, en este módulo podemos consultar la evaluación creada por el profesor para agregar observaciones o modificar el recurso educativo evaluado. A continuación se presentan los requerimientos de este módulo.

Mostrará un listado con los datos de las evaluaciones: Id de recurso, Título, Observaciones, Plan/Elemento del Programa de estudios, Fecha de evaluación, Usuario que evaluó el recurso, último usuario que dio seguimiento a la evaluación, Dictamen, Calificación global y Estado de seguimiento.

Permitirá buscar la búsqueda con los siguientes datos: Título, Id, Usuario que evaluó, Estatus de seguimiento, Plan de estudios, Calificación global y Dictamen.

Permitirá exportar la información consultada a Excel.

Permitirá consultar los resultados de la evaluación(rúbrica) para registrar observaciones y asignar un estatus de atendida o pendiente a la evaluación.

Permitirá exportar a Excel las respuestas de las evaluaciones mostradas en el listado.

El acceso a este módulo deberá ser restringido sólo a los usuarios con permiso de administración de recursos educativos.

3. **Seguimiento de recursos sugeridos:** Este módulo permite el registro de un recurso educativo a través de una sugerencia realizada por algún usuario de la comunidad escolar desde el portal. Adicionalmente se puede eliminar la sugerencia.

Módulo de Reportes

1. **Reporte Listado de recursos educativos*:** Este módulo permite consultar un listado con la información de los recursos educativos. A continuación se presentan los requerimientos de este módulo.

Permitirá la búsqueda con los siguientes datos: Valoraciones, Autor, Usuario que creó/importó, Plan de estudios, Formato, Categoría, Estado del recurso y Nivel de acceso.

Permitirá elegir la información que se mostrará en el listado la cual puede ser: Tipo de formato, Detalle de la asociación, Fecha de creación, Usuario que creó/importó el recurso, Usuario que realizó la última modificación, Estado de publicación.

Permitirá ordenar el listado de coincidencias por cualquiera de sus columnas a excepción de la de “Detalle de la asociación”. Permitirá exportar la información consultada a Excel.

El acceso a este módulo deberá ser restringido sólo a los usuarios con permiso de consultar el listado de recursos educativos.

- 2. Reporte estadísticas de recursos educativos*:** Este módulo permite generar un reporte con el conteo de recursos educativos, según los datos de búsqueda ingresados. A continuación se presentan los requerimientos de este módulo.

La información desplegada en la tabla con los resultados de la búsqueda será: Ruta de vinculación, Ciclo, Recursos asociados directamente y Recursos asociados directamente y niveles inferiores.

Permitirá la búsqueda con los siguientes datos: Los criterios de búsqueda que se pueden seleccionar son: Plan de estudios, Formato, Categoría, Estado y Fuente del recurso, Estado de publicación y Nivel de acceso.

Permitirá exportar la información consultada a Excel.

El acceso a este módulo deberá ser restringido sólo a los usuarios con permiso de consultar estadísticas de recursos educativos.

- 3. Reporte estatus de URL*:** Este módulo permite consultar un listado de recursos educativos para conocer el estatus de la URL del recurso. A continuación se presentan los requerimientos de este módulo.

La información desplegada en la tabla con los resultados de la búsqueda será: Id, Categoría, Estado de publicación, Estatus, Usuario que realizó la última modificación y URL.

Permitirá la búsqueda con los siguientes datos: Los criterios de búsqueda que se pueden seleccionar son: Plan de estudios, Categoría, Estado de publicación y Tipo de respuesta de URL.

El sistema realizará de manera automática la verificación de los estatus de URL de los recursos publicados.

Se mostrará en el listado la fecha de la última vez que el sistema realizó la revisión de URL.

Permitirá exportar la información consultada a Excel.

El acceso a este módulo deberá ser restringido sólo a los usuarios con permiso

de consultar estatus de URL.

4. **Reporte de evaluaciones de recursos:** Este módulo permite la consulta de información relacionada a las evaluaciones de los recursos educativos.
5. **Reporte de recursos sugeridos:** Este módulo permite la consulta de información relacionada con los recursos sugeridos.
6. **Reporte de seguimiento:** Este módulo permite la consulta de información de recursos educativos y planes de estudios.

Asociación de elementos

Este módulo permite la creación y eliminación de asociaciones entre elementos y el plan de estudios, esta funcionalidad está presente en los módulos: administración de recursos educativos, administración de planes de estudios y administración del programa de estudios. A continuación se presentan los requerimientos:

Permitirá agregar y eliminar asociaciones del elemento seleccionado.

Presentará las siguientes secciones para las asociaciones dependiendo del elemento del cual se administren las asociaciones:

Plan o elemento del programa de estudios: “Lo que ya debo saber”, “Para Saber Más” y “Recursos educativos”.

Recursos educativos: “Lo que ya debo saber”, “Para Saber Más” y “Elementos del plan de estudios”.

Adicionalmente las sección de recursos educativos será dividida en las categorías: Autoevaluación, Recursos para aprender y Recursos para el profesor

Permitirá buscar elementos por título e id para asociar.

Permitirá mostrar el detalle de la información que se encuentra asociada de: recursos educativos, planes de estudios y recursos educativos.

El acceso a este módulo deberá ser restringido sólo a los usuarios con los permisos: administración de recursos educativos y administración de planes de estudios.

2.1.2. Requerimientos portal RUA IES

Sugerencia de recursos educativos: Este módulo permite a los usuarios de la comunidad escolar de tipo profesor sugerir recursos educativos para que sean incorporados dentro de la RUA IES. El usuario que sugiera el recurso no debe de completar toda la información requerida para la sugerencia del recurso, únicamente debe de proporcionar la fuente del recurso educativo.

Registro de Evaluación de recursos educativos: Este módulo permite a los usuarios de la comunidad escolar de tipo profesor evaluar los recursos publicados. Los aspectos a evaluar son: autoridad, credibilidad, relevancia, pertinencia, velocidad, navegabilidad, usabilidad y legibilidad.

Buscador de recursos educativos: Este módulo permite la consulta de los recursos educativos publicados. Existen dos tipos de búsqueda dentro del portal.

- **Búsqueda básica:** Permitirá ingresar texto para obtener resultados de recursos educativos que coincidan con lo que se haya ingresado.
- **Búsqueda avanzada:** Permitirá ingresar texto además de acotar la búsqueda de recursos educativos por medio de criterios como Categoría, Formato y Plan de estudios.

Navegación por plan de estudio: Este módulo permite navegar mediante la estructura correspondiente a un plan de estudios para ubicar recursos educativos y elementos del programa de estudios. El sistema presentará los elementos del programa de estudios y recursos educativos agrupados en las categorías:

- “Lo que debo saber”
- “Recursos para aprender”
- “Autoevaluación”
- “Recursos para el profesor”
- “Para saber más”

Consulta de ficha técnica: Este módulo permite consultar la información de un recurso educativo dentro del portal. **Administrar Página del Profesor:** Este módulo permite personalizar la página de un profesor por medio de la creación edición y eliminación de información.

Registrar comentarios y sugerencias: Este módulo permite enviar comentarios o sugerencias con ayuda de una captura de pantalla del portal de la RUA IES.

Historial de Evaluaciones del Profesor: Este módulo permite a un profesor consultar las evaluaciones de recursos educativos que ha realizado.

2.1.3. Requerimientos no funcionales RUA IES

- El sistema debe utilizar el protocolo de transferencia de hipertexto (HTTP, Hyper-Text Transfer Protocol), usado en cada transacción de la Web (WWW).
- El sistema será desarrollado con el lenguaje de programación PHP con el uso del framework CakePHP.
- La información será almacenada en claro en la base de datos, excepto la contraseña que se guardará cifrada.

- El sistema debe llevar, mediante una bitácora de movimientos, un control de los accesos al sistema, así como también el registro de los movimientos realizados:
 1. Inicio de sesión.
 2. Cierre de sesión.
 3. Altas, bajas y cambios de los siguientes términos:
 - a) Recursos educativos
 - b) Usuarios
 - c) Planes y programas de estudios
 - d) Asociaciones entre elementos (recursos educativos y planes y programas de estudios)

- El portal debe presentar una interfaz adaptable con lo que pueda verse de manera clara en distintos dispositivos.

Con base en los requerimientos antes descritos, podemos observar que una de las funcionalidades de la RUA IES que se presenta en distintos módulos es la búsqueda de los distintos elementos presentes en la RUA IES en el programa de estudios, una actividad que se busca que sea una de las principales características de la RUA IES, ya permitirá al usuario encontrar los recursos que necesita de acuerdo con el Plan de estudios que sea de su interés.

2.2. Diagrama general de casos de uso

En la figura 2.1 se presenta el diagrama general de Casos de Uso de los casos de usos en los cuales participé, con los actores que se identificaron por medio de los perfiles encontrados en los distintos requerimientos funcionales. El diagrama general de casos de uso completo se encuentra en la documentación de RUA[3].

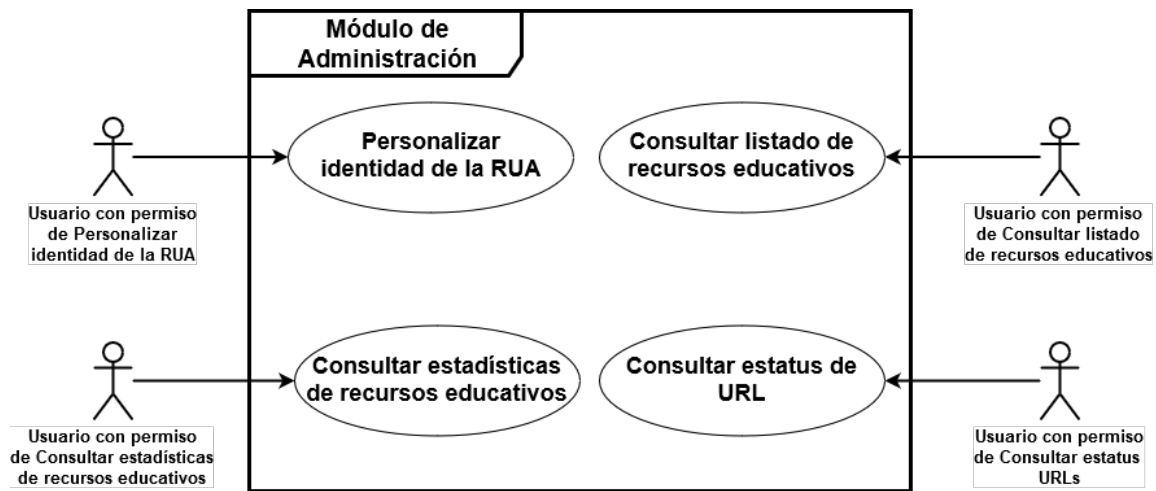


Figura 2.1: Diagrama general de casos de uso módulo de administración

2.3. Funcionalidades de los módulos: listado recursos educativos y seguimiento de evaluaciones

En esta sección se describen las funcionalidades que debe tener el módulo de *listado de recursos educativos y seguimiento de evaluaciones*. Estas funcionalidades se obtuvieron a partir de los requerimientos antes descritos y para los casos de uso: listado de recursos educativos[5] y seguimientos de evaluaciones de recursos educativos[4]. Ambos casos de uso se encuentran en la sección de Anexos de este documento.

2.3.1. Funcionalidades del módulo: listado de recursos educativos

Las funcionalidades identificadas con base en los requerimientos son:

1. Consultar listado de recursos educativos.
2. Buscar recursos educativos a través de los criterios de búsqueda.
3. Exportar a Excel los recursos educativos presentes en el listado.

En la figura 2.2 presentamos un diagrama con las funcionalidades de este módulo.

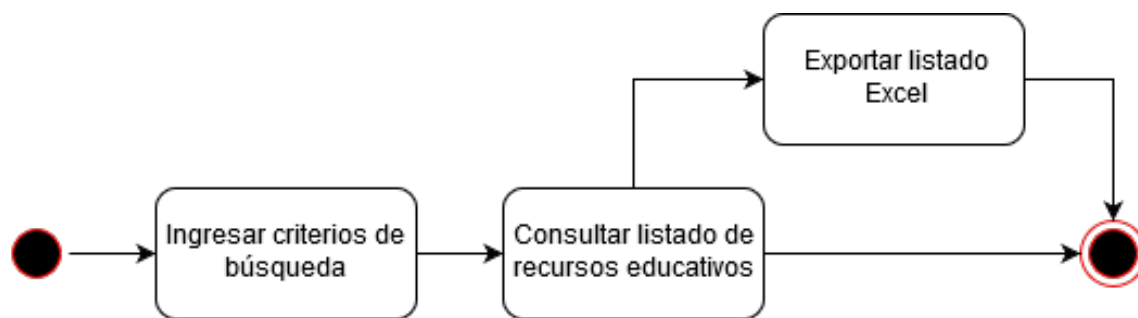


Figura 2.2: Diagrama de procesos listado de recursos educativos

2.3.2. Funcionalidades del módulo: seguimiento de evaluaciones

Las funcionalidades identificadas con base en los requerimientos son:

1. Consultar listado de evaluaciones.
2. Buscar evaluaciones de recursos educativos a través de los criterios de búsqueda.
3. Exportar a Excel el listado de evaluaciones.
4. Consultar el resultado de la evaluación(rúbrica).
5. Agregar observaciones para atender la evaluación del recurso educativo.
6. Exportar a Excel el resultado de las evaluaciones presentes en el listado.

En la figura 2.3 presentamos un diagrama con las funcionalidades de este módulo.

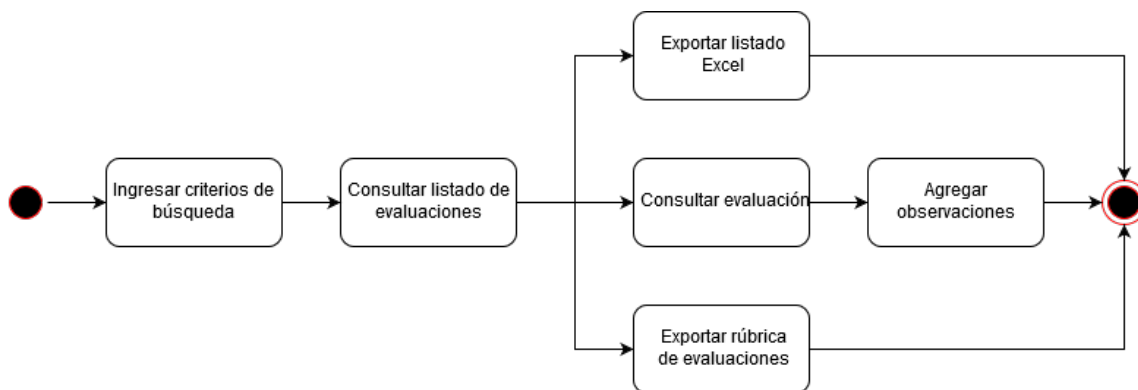


Figura 2.3: Diagrama de procesos seguimiento de evaluaciones de recursos educativos

2.4. Reglas de asociación de elementos

Como mencionamos con anterioridad la búsqueda de elementos en el plan de estudios dentro de RUA es una funcionalidad presente en distintos módulos del sistema, por lo cual en esta sección describiremos las reglas de asociación entre el plan de estudios y los recursos educativos.

Dentro de RUA hay tres elementos que pueden ser asociados, estos son:

1. **Nodos de primer nivel:** Corresponden al nombre de algún plan de estudios. En la relación este elemento representa el nodo de inicio desde el cual se expandirá el árbol del programa de estudios.
2. **Nodos:** Corresponden a los distintos niveles que detallan el plan de estudios y dan forma al programa de estudios. Estos pueden corresponder a: años, semestres, materias, asignaturas, temas, o cualquier nivel que se requiera de acuerdo al plan de estudios que se desea especificar. Los nodos deben estar asociados a un plan de estudios solo como hijos o cualquier otro nodo existente.
3. **Recursos educativos:** Los recursos educativos pueden estar asociados tanto a Nodos de primer nivel, Nodos u otros recursos educativos. Los recursos educativos corresponden a las hojas dentro del programa de estudios y no pertenecen a la estructura del programa de estudios.

En la figura 2.4 presentamos las relaciones que se pueden presentar para conformar el programa de estudios.

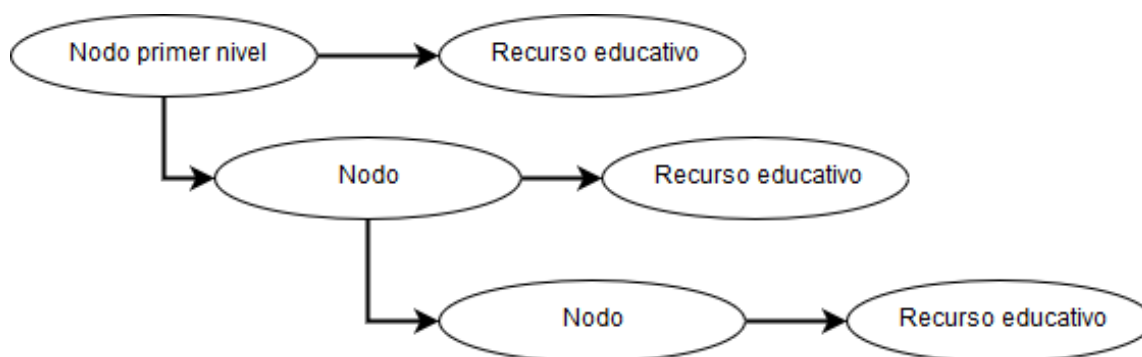


Figura 2.4: Elementos y asociaciones

Estos tres elementos tienen distinta información, sin embargo la información que comparten es almacenada en la misma tabla dentro de la base de datos. Dentro de RUA estos elementos pueden tener cuatro tipos de relaciones. En la figura 2.5 observamos la abstracción de elementos y asociaciones que se pueden dar entre estos para realizar la búsqueda dentro de los planes de estudios.

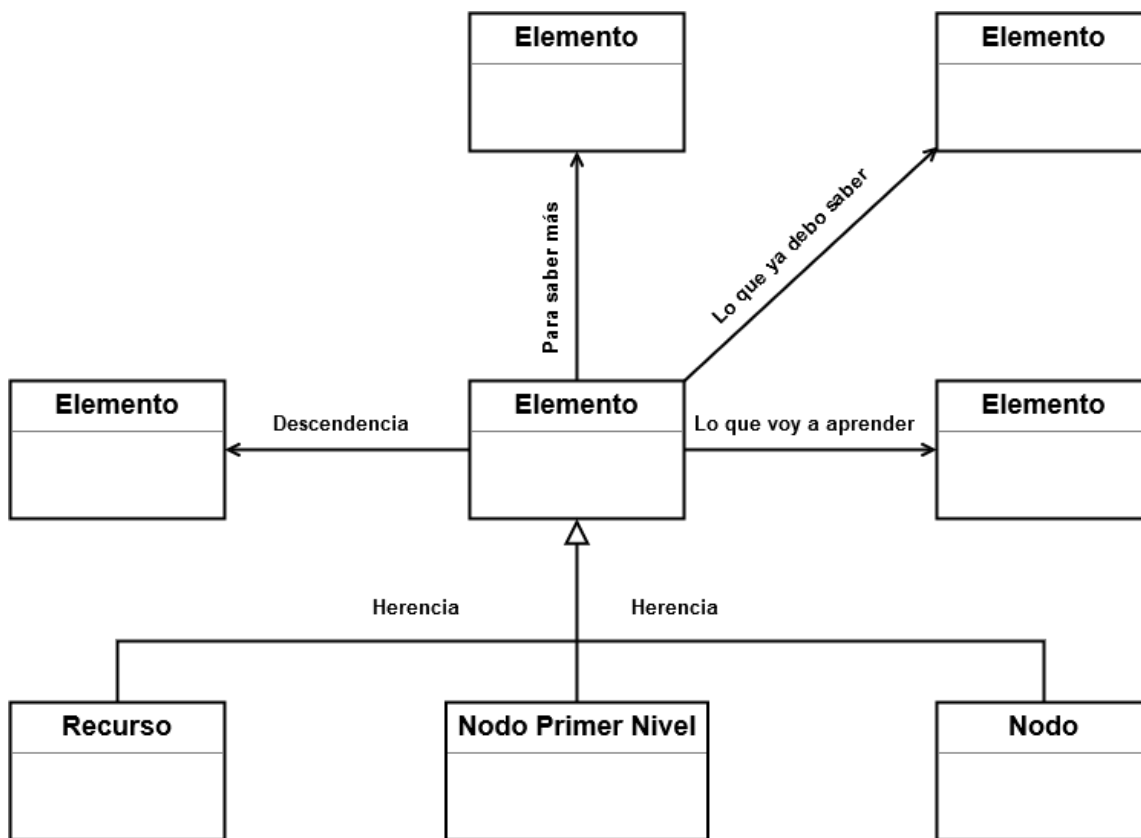


Figura 2.5: Relaciones entre elementos

A continuación describiremos los cuatro tipos distintos de relación que existen entre los elementos de la RUA.

1. **Lo que ya debo saber** – Esta relación representa los elementos que se recomiendan conocer para acceder a algún contenido; la relación puede darse entre cualquier tipo de elemento.
2. **Lo que voy a aprender** – Esta relación representa los conocimientos que serán adquiridos a través del elemento; la relación puede darse entre cualquier tipo de elemento.
3. **Para saber más** – Esta relación presenta sugerencias para profundizar en el tema del elemento con el que se asocia; la relación puede darse entre cualquier tipo de elemento.
4. **Descendencia** – Esta relación es la que forma la estructura del programa de estudios y puede darse **solo** entre elementos de tipo **Nodo de primer nivel** y **Nodos**, donde

los Nodos de primer nivel solo pueden presentarse como raíz dentro de la estructura de gráfica del programa de estudios bajo esta relación. Teniendo únicamente un elemento de tipo nodo primer nivel en cada programa de estudios.

2.5. Arquitectura de RUA IES

En esta sección con base en los requerimientos antes descritos presentaremos la arquitectura de RUA IES. La RUA IES es un sitio web que es organizado en dos módulos, como se ha mencionado en la especificación de requerimientos; ambos módulos: Portal y módulo de administración; se encuentran conectados a la misma base de datos con la diferencia en que el portal puede verse como un sitio que solo hace lecturas de la base de datos y en el módulo de administración se realiza la modificación y creación de los datos. Las tecnologías usadas para el desarrollo de este proyecto son:

PHP 5.1	CakePHP 2.5	PotgreSQL 9.2
Apache 2.4	HTML 5	CSS 3
jQuery 1.9	Bootstrap 3	BootstrapValidator

En la figura 2.6 mostramos la arquitectura de RUA IES:

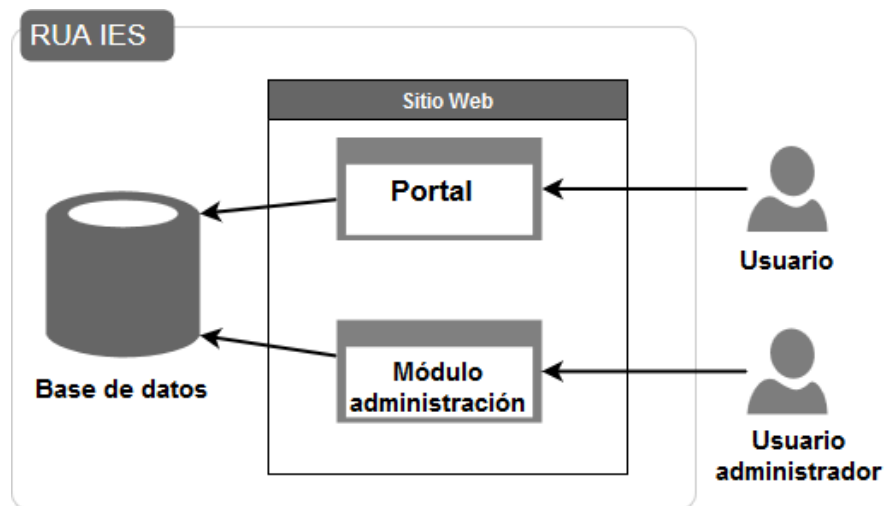


Figura 2.6: Arquitectura RUA IES

2.5.1. Base de datos

El sitio de RUA cuenta con una base de datos PostgreSQL, en esta sección se describen las entidades utilizadas en los módulos que se han venido describiendo:

1. Listado de recursos educativos: Las tablas utilizadas en esta sección son:

- **recurso:** Es la tabla donde se almacena la información del recurso, en este módulo nos permite realizar la búsqueda por: Categoría (`categoria_id`), Estado del recurso (`estado_id`), Fuente del recurso (`recur_tipo_recurso`) y Nivel de acceso (`recur_nivel_acceso`). Adicionalmente se presentan en el listado los campos: Id Recurso (`elemento_id`) y Fuente del recurso educativo (`recur_url`); solo cuando la fuente del recurso es una URL.
- **elemento:** Es la tabla donde se almacenan los datos principales del recurso; en este módulo nos permite realizar la búsqueda por: Texto (`tsv_nombre`, `tsv_descripcion`), Autor (`tsv_autor`), Usuario que creó/importó el recurso (`usuario_id`), Usuario que realizó la última modificación (`usuario_id_modificacion`), Estado de publicación (`elem_estatus_publicacion`) y Fecha de creación (`elem_fecha_creacion`). Adicionalmente se presentan en el listado los campos: Título del recurso (`elem_nombre`), Fecha de creación (`elem_fecha_creacion`) y estado de publicación (`elem_estatus_publicacion`).
- **usuario:** Es la tabla donde se almacenan los usuarios administradores del sistema, estos usuarios son lo que pueden crear, editar y eliminar recursos educativos; en este módulo permite presentar en el listado los campos: Usuario que creó/importó el recurso y Usuario que realizó la última modificación (`usu_nombre`, `usu_primer_apellido`, `usu_segundo_apellido`).
- **formato:** Es la tabla transitiva que nos permite identificar el Tipo de formato del recurso (`formato_id`).
- **tipo_formato:** Es una tabla contiene los valores de los tipos de formato que puede tener un recurso, en este módulo nos permite presentar: la opción de búsqueda y en el listado el campo: Tipo de formato (`tipo_form_nombre`).
- **estado:** Es la tabla que contiene los valores de los estados que puede tener un recurso, en este módulo nos permite presentar la opción de búsqueda por el campo Estado del recurso (`estado_nombre`).
- **categoría:** Es la tabla que contiene los valores de las categorías que puede tener un recurso, en este módulo nos permite presentar: la opción de búsqueda y en el listado el campo: Estado del recurso (`cat_nombre`).

En la figura 2.7 presentamos el esquema de la base de datos únicamente con las entidades y relaciones usadas en este módulo.

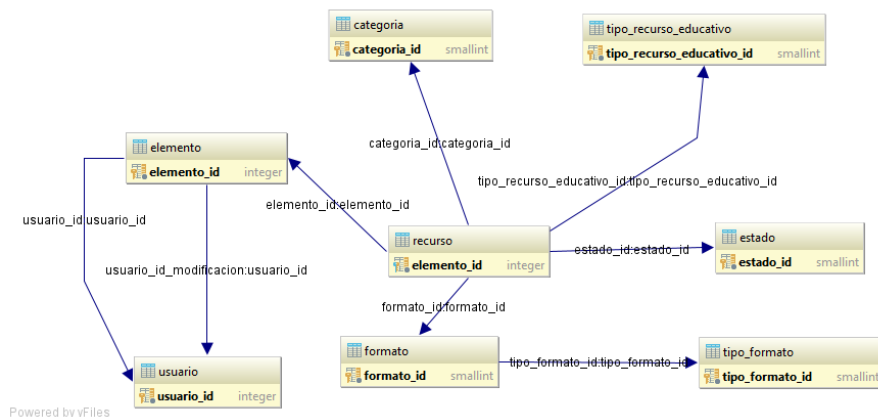


Figura 2.7: Esquema de la base de datos listado de recursos educativos

2. **Seguimiento de evaluación de recursos educativos:** Las tablas utilizadas en esta sección, adicionales a las descritas con anterioridad son:

- **evaluacion:** Es la tabla donde se almacena la evaluación realizada por un usuario de la comunidad escolar de tipo profesor, en este módulo nos permite realizar la búsqueda por los campos: Usuario que evaluó el recurso (usuario_comunidad_id), Último usuario que dio seguimiento a la evaluación (usuario_id), Estado de seguimiento (eval_estatus_atencion), Calificación global (eval_calificacion), Fecha de evaluación (eval_fecha_evaluacion) y Dictamen (eval_dictamen). Adicionalmente se presentan en el listado los campos: Id (evaluacion_id), Estado de seguimiento (eval_estatus_atencion), Calificación global (eval_calificacion) y Dictamen (eval_dictamen).
- **usuario:** En este módulo permite presentar en el listado la información de Último usuario que dio seguimiento a la evaluación (usu_nombre, usu_primer_apellido, usu_segundo_apellido).
- **usuario_comunidad:** En este módulo permite presentar en el listado la información de Usuario que evaluó el recurso (usu_com_nombre, usu_com_primer_apellido, usu_com_segundo_apellido).
- **recurso:** En este módulo permite mostrar en el listado y buscar por medio del campo Id del recurso (elemento_id).
- **elemento:** En este módulo permite mostrar en el listado y buscar por medio del campo Título del recurso (elem_nombre, tsv_titulo y tsv_descripcion).
- **evaluacion_pregunta:** En esta tabla se almacenan las respuestas que el profesor puso en cada pregunta de la evaluación del recurso. En este módulo nos

permite mostrar las respuestas (eval_pre_respuesta) de cada pregunta en la funcionalidad de consulta de la evaluación.

- **pregunta:** En esta tabla se almacenan las distintas preguntas que contiene una evaluación. En este módulo nos permite listar las preguntas (preg_pregunta) en la funcionalidad de consulta de evaluación.
- **seccion_pregunta:** Es una tabla que contiene las secciones a las que pertenece cada pregunta. En este módulo nos permite agrupar las preguntas por secciones (sec_preg_titulo) en la funcionalidad de consulta de evaluación.

En la figura 2.8 presentamos el esquema de la base de datos únicamente con las entidades y relaciones usadas en este módulo.

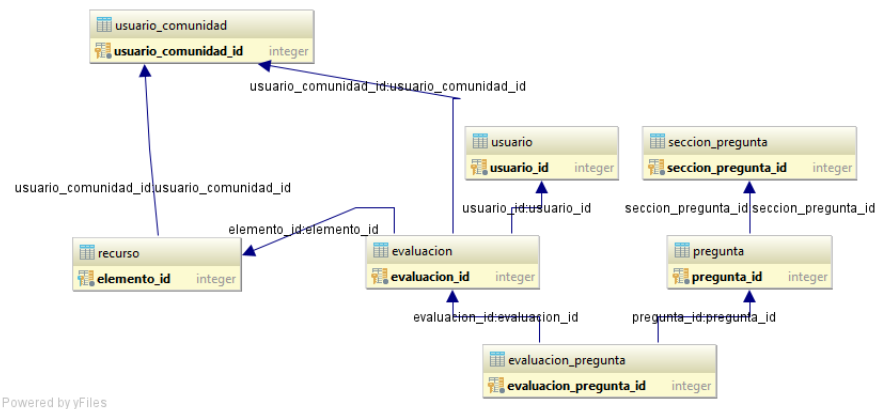


Figura 2.8: Esquema de la base de datos seguimiento de evaluaciones

2.5.2. Búsqueda en el programa de estudios de recursos educativos

Con los diagramas anteriores describimos la mayoría de los datos necesarios para la implementación de los módulos descritos, sin embargo en ambos módulos faltó describir la búsqueda a través del plan de estudios. A continuación describiremos esta búsqueda y las tablas de la base de datos que nos ayudarán a lograrla.

La búsqueda de elementos a través del plan de estudios inicia con un elemento de tipo nodo de primer nivel, el cual corresponde a un plan de estudios. una vez que el usuario selecciona un plan de estudios sobre el cual quiere realizar la búsqueda, el sistema muestra un listado con los elementos de tipo nodo asociados directamente al plan de estudios a través de la relación de tipo *descendencia*.

En caso que los nodos anteriores, tengan más relaciones con elementos de tipo nodo a través

de la relación de tipo *descendencia*, el usuario puede expandir cualquiera de estos nodos para listar los elementos asociados; y repetir esta acción hasta que el nodo no tenga más relaciones con otros nodos. A esta característica la llamaremos **exploración del árbol del programa de estudios**.

Adicionalmente a la característica antes descrita, para realizar la búsqueda el usuario debe seleccionar los nodos a partir de los cuales quiere realizar la búsqueda, el usuario puede seleccionar que se busque a partir del plan de estudios directamente.

Una vez que el usuario seleccionó los elementos en el árbol del programa de estudios la búsqueda se realiza encontrando todos los recursos educativos asociados directamente a los elementos seleccionados y de los nodos descendentes del árbol seleccionado.

En la figura 2.9 mostramos la exploración del árbol del programa de estudios correspondiente al módulo de administración.

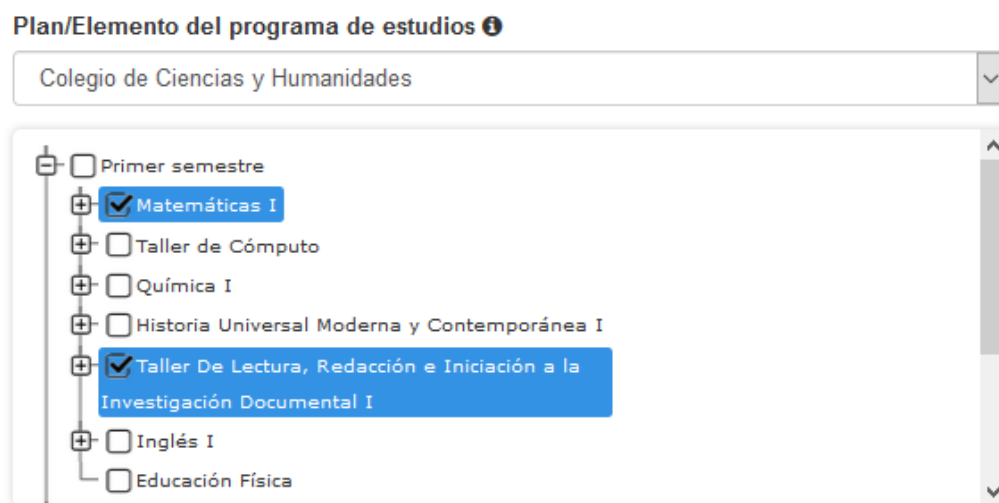


Figura 2.9: Exploración del programa de estudios

A continuación describiremos las tablas que nos permiten realizar la búsqueda y exploración a través de programa de estudios:

- **elemento**: Es la tabla en la cual se almacenan los campos comunes de los elementos que conforman el programa de estudios. Adicionalmente esta tabla contiene el identificador (`elemento_id`) de cada elemento, el cual nos servirá para guardar las relaciones del elemento en la tabla **relacion**.
- **tipo_elemento**: Es la tabla que contiene los valores de tipo elemento que puede tener un elemento; los tipo de elemento que existen son: nodo primer nivel (Plan de estudios), nodos (Elemento del programa de estudios) y recursos educativos.

- **nodo_primer_nivel:** Es la tabla donde se almacenan los datos de los planes de estudios que no comparte con los otros elementos.
- **nodo:** Es la tabla donde se almacenan los datos de los elementos del programa de estudios que no comparten con los otros elementos.
- **recurso:** Es la tabla donde se almacenan los datos del recurso educativo que no comparte con los otros elementos.
- **relacion:** Es la tabla donde se almacenan las relaciones entre los elementos. Estas relaciones son dirigidas que contienen un elemento de origen (elemento_id_origen) y un elemento destino (elemento_id_destino). Adicionalmente las asociaciones pueden ser de cuatro tipos distintos.
- **tipo_relacion:** Es la tabla que contiene los valores de tipo relación que puede tener una relación; los tipos de relación que existen son: Descendencia, Lo que ya debo de saber, Lo que voy a aprender y Para saber más.

En la figura 2.10 presentamos el esquema de la base de datos únicamente con las entidades y relaciones usadas en esta funcionalidad.

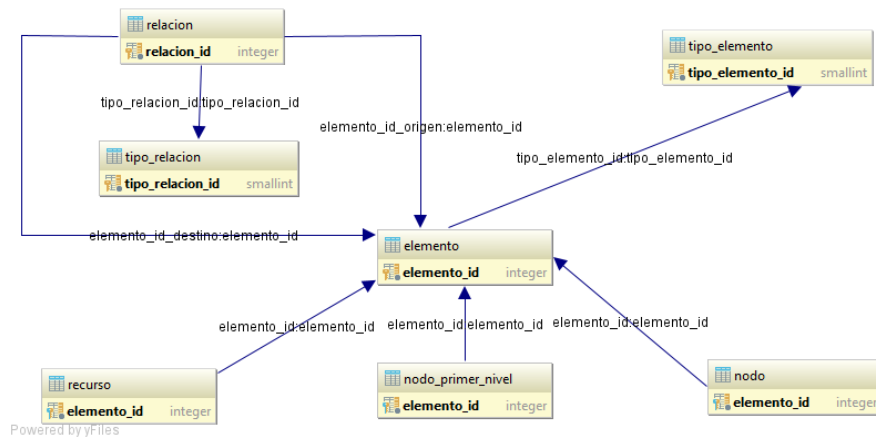


Figura 2.10: Esquema de la base de datos búsqueda en el programa de estudios

2.6. Estructura RUA IES

En esta sección se listan los componentes del proyecto de RUA IES, para lograr esto comenzaremos con describir los componentes del framework utilizado en el desarrollo del sitio para posteriormente describir los componentes de la RUA IES.

2.6.1. Componentes de CakePHP

En una aplicación básica de CakePHP podemos observar tres componentes principales que son la base de una aplicación, a continuación se describen:

1. **Cake** Es el núcleo del *framework*, en esta carpeta se encuentran los componentes, objetos y funciones que permiten el funcionamiento de CakePHP.
2. **App** Aquí encontramos los componentes que serán desarrollados para cada aplicación, en esta carpeta se encuentran las tres capas del patrón de diseño MVC, que son el Modelo la Vista y el Controlador, así como archivos de configuración para el funcionamiento de la aplicación.
3. **Webroot** En esta carpeta se encuentran los archivos que son accesibles en internet, tales como imágenes, hojas de estilos y archivos JavaScript.

En la figura 2.11 presentamos el diagrama de componentes de una aplicación típica de CakePHP.

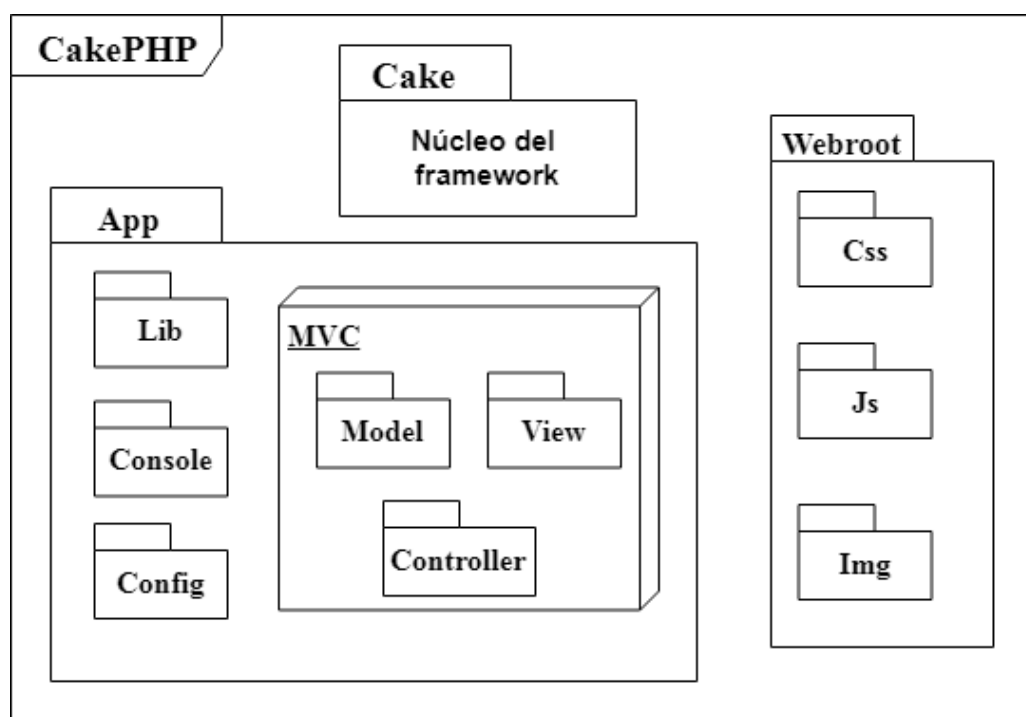


Figura 2.11: Componentes CakePHP

Adicionalmente a las carpetas antes descritas, como podemos observar en la figura 2.11 dentro de la carpeta de App se encuentran las carpetas: Model, View y Controller; dentro

de estas carpetas se encontrarán las clases correspondientes para la implementación de la aplicación, a continuación describiremos con mayor detalle estas tres carpetas para entender la adaptación del patrón de diseño MVC dentro de CakePHP:

1. **Model:** Dentro de esta carpeta se encontraran los Modelos de la aplicación. Los **Modelos** dentro de CakePHP representan una tabla de la base de datos y se encargan del acceso a los datos de esta.
2. **View:** Dentro de esta carpeta se encuentran las Vistas de la aplicación. Las **Vistas** corresponden a la respuesta de acuerdo a la petición realizada. Las Vistas dentro de CakePHP están asociadas a la acción de un Controlador.
3. **Controller:** Dentro de esta carpeta se encuentran los Controladores de la aplicación. Los **Controladores** se encargan de manejar los datos enviados en la petición. Cada petición del usuario se encuentra asociada a un método de un controlador los cuales llamamos acciones.

2.6.2. Estructura RUA IES

Como se mencionó con anterioridad la RUA IES se divide en dos módulos, el módulo de administración y el módulo del portal. Estos módulos son dirigidos a distintos usuarios, ya que el módulo del portal es totalmente público sin embargo el módulo de Administración solo es para usuarios Administradores.

Debido a la interacción que existe entre ambos módulos estos comparten distintas características y funcionalidades por lo cual para evitar la redundancia en el código de la aplicación se optó por crear tres proyectos en CakePHP; uno para cada módulo principal y un tercero para colocar los componentes que son compartidos entre ambos módulos. En la figura 2.12 se presenta la estructura del proyecto de la RUA IES. A continuación describiremos los componentes que la conforman.

- **Cake** Corresponde al núcleo de CakePHP que es compartida entre los tres proyectos.
- **Webroot** son los archivos accesibles en la red de este proyecto, este a su vez se divide entre los tres proyectos de CakePHP que componen a la RUA IES, el admin, portal y rua.
- **SRC** Corresponde a la carpeta app descrita anteriormente en ella se encuentra la implementación de los distintos sub-módulos de cada aplicación, de igual manera se divide entre las carpetas: admin, portal y rua.
 - **Admin** En estas carpetas se encuentran los objetos, funciones, componentes, bibliotecas y archivos que componen la implementación de los distintos sub-módulos que componen el módulo de administración. Este módulo permite la

creación, actualización, catalogación, eliminación y actualización del contenido que podemos encontrar en la RUA IES.

- **Portal** En estas carpetas se encuentran los objetos, funciones, componentes, bibliotecas y archivos que componen la implementación de los distintos submódulos que componen el módulo de administración. Este módulo la búsqueda y consulta de los distintos elementos que se encuentren catalogados dentro de la RUA IES.
- **RUA** En estas carpetas se encuentran los objetos, funciones, componentes, bibliotecas y archivos que son compartidos entre las aplicaciones: **Admin** y **Portal**.

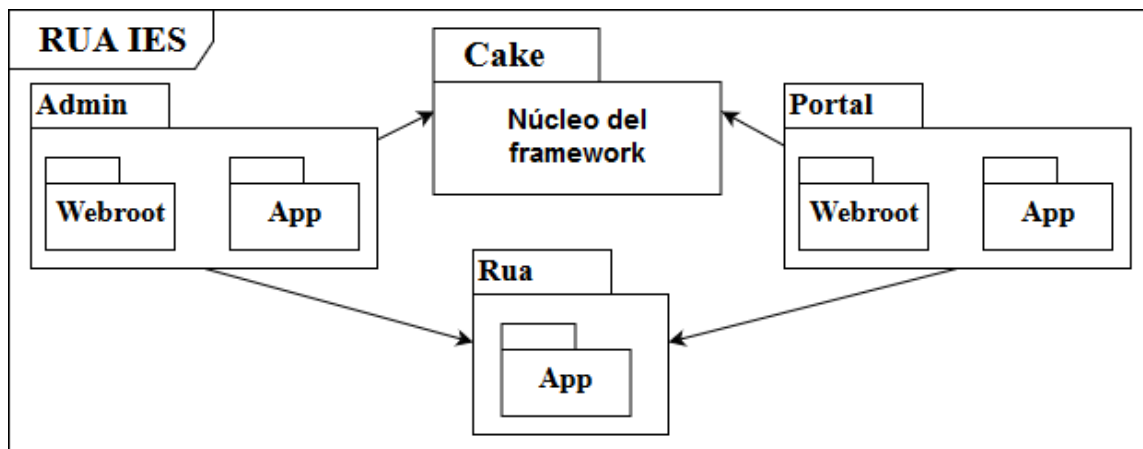


Figura 2.12: Componentes RUA IES

Con esta estructura de la RUA IES obtenemos la división del módulo de administración y el módulo del portal y con el proyecto de rua tenemos los componentes compartidos entre ambos módulos para de esta manera evitar la redundancia del código entre ambos módulos y mantener divididos ambos sitios.

2.7. Vista estática RUA IES

En esta sección detallaremos las clases, y sus relaciones, que componen la implementación del seguimiento de evaluaciones de recursos educativos y el listado de recursos educativos.

2.7.1. Diagrama de clases listado de recursos educativos

La implementación del módulo de listado de recursos educativos se compone de tres componentes principales siguiendo el patrón de diseño MVC, a continuación detallaremos los componentes de este módulo bajo la adaptación de CakePHP del patrón de diseño.

1. **ReporteListadoRecursoController:** Corresponde al controlador del módulo, sus métodos corresponden a las acciones que puede realizar el usuario dentro de este módulo.
 - a) **index:** Corresponde a la acción de mostrar los criterios de búsqueda de las evaluaciones, en esta acción se cargan además los elementos necesarios para generar el listado de los recursos educativos.
 - b) **listar:** Corresponde a la acción que genera el listado de los recursos educativos, adicionalmente filtra las evaluaciones de acuerdo a los campos proporcionados por el usuario.
 - c) **generarReporteExcel:** Corresponde a la acción que genera un documento Excel con los datos de los recursos educativos que se presentan en el listado.
2. **GenerarReporte:** Corresponde al modelo, esta clase se asocia con la tabla *evaluacion* de la base de datos. Este modelo se encarga de validar los datos guardados en el seguimiento de evaluaciones y brinda las funciones necesarias para obtener los datos necesarios por las acciones del controlador.
3. **Vistas:** Las vistas utilizadas en este módulo son: index y listar; asociadas a las acciones correspondientes del controlador; estas vistas utilizan un archivo *javascript* para mejorar la interacción del usuario con el sitio.

En la figura 2.13 presentamos las clases utilizadas en la implementación de este módulo.

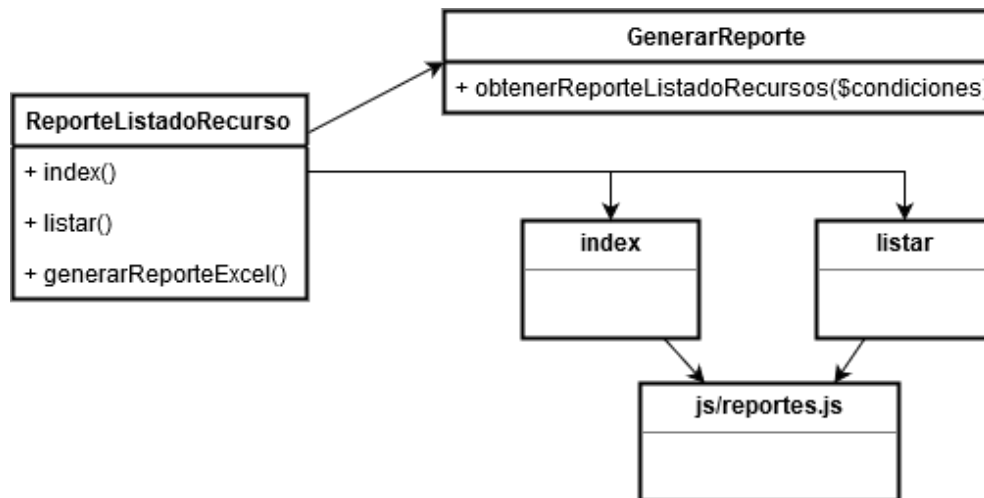


Figura 2.13: Diagrama de clases Listado de recursos educativos

2.7.2. Diagrama de clases seguimiento de evaluaciones de recursos educativos

La implementación del módulo de listado de recursos educativos se compone de tres componentes principales siguiendo el patrón de diseño MVC, a continuación detallaremos los componentes de este módulo bajo la adaptación de CakePHP del patrón de diseño.

1. **EvaluacionesController:** Corresponde al controlador del módulo, sus métodos corresponden a las acciones que puede realizar el usuario dentro de este módulo.
 - a) **index:** Corresponde a la acción de mostrar los criterios de búsqueda de las evaluaciones, en esta acción se cargan además los elementos necesarios para generar el listado de las evaluaciones.
 - b) **listar:** Corresponde a la acción que genera el listado con las evaluaciones, adicionalmente filtra las evaluaciones de acuerdo a los campos proporcionados por el usuario.
 - c) **reporteListado:** Corresponde a la acción que genera un documento Excel con los datos de las evaluaciones que se presentan en el listado.
 - d) **reporteRubrica:** Corresponde a la acción que genera un documento Excel donde se encuentran los resultados de cada sección de la evaluación
 - e) **consultar:** Corresponde a la acción que permite agregar observaciones a las evaluaciones hechas, adicionalmente presenta los resultados de cada sección de la evaluación del recurso.
 - f) **guardarSeguimiento:** Corresponde a la acción que guarda las observaciones realizadas por el administrador en la acción de consulta.
2. **Evaluacion:** Corresponde al modelo, esta clase se asocia con la tabla *evaluacion* de la base de datos. Este modelo se encarga de validar los datos guardados en el seguimiento de evaluaciones y brinda las funciones necesarias para obtener los datos necesarios por las acciones del controlador.
3. **Vistas:** Las vistas utilizadas en este módulo son: *index*, *consultar* y *listar*; asociadas a las acciones correspondientes del controlador; estas tres vistas utilizan un archivo *javascript* para mejorar la interacción del usuario con el sitio.

En la figura 2.14 presentamos las clases utilizadas en la implementación de este módulo.

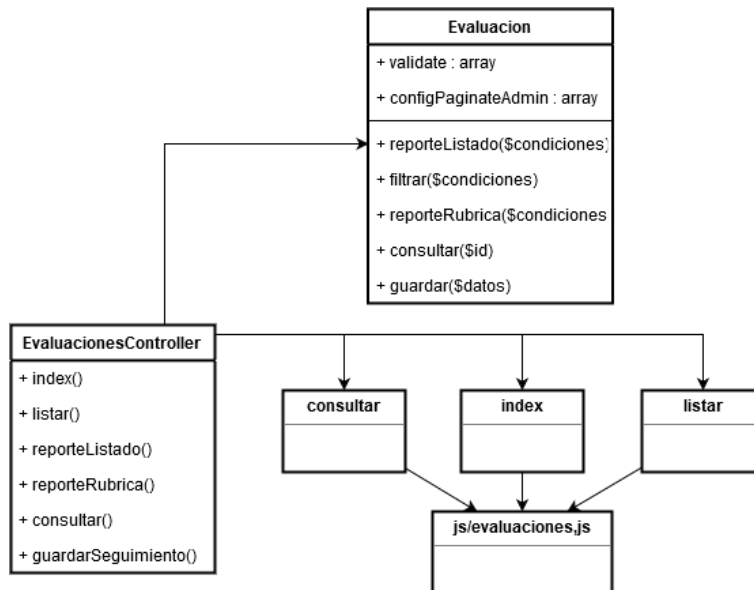


Figura 2.14: Diagrama de clases Seguimiento de evaluaciones

2.8. Vista dinámica RUA IES

En esta sección veremos la interacción de los distintos componentes, de los módulos: listado de recursos educativos y seguimiento de evaluaciones; para entender la implementación realizada para cumplir con los requerimientos de cada módulo.

2.8.1. Diagrama de secuencia del módulo: listado de recursos educativos

En esta sección describiremos las funcionalidades antes mencionadas del módulo de listado de recursos educativos. Describiremos la interacción del usuario con el sistema de cada funcionalidad del módulo:

1. Ingresar “Listado de recursos educativos”
 - a) Esta funcionalidad inicia cuando el usuario selecciona en el menú del sistema Listado de Recursos educativos en la sección de Reportes.
 - b) El sistema carga los distintos catálogos necesarios para presentar los criterios de búsqueda.
 - c) El sistema muestra los criterios de búsqueda del reporte.

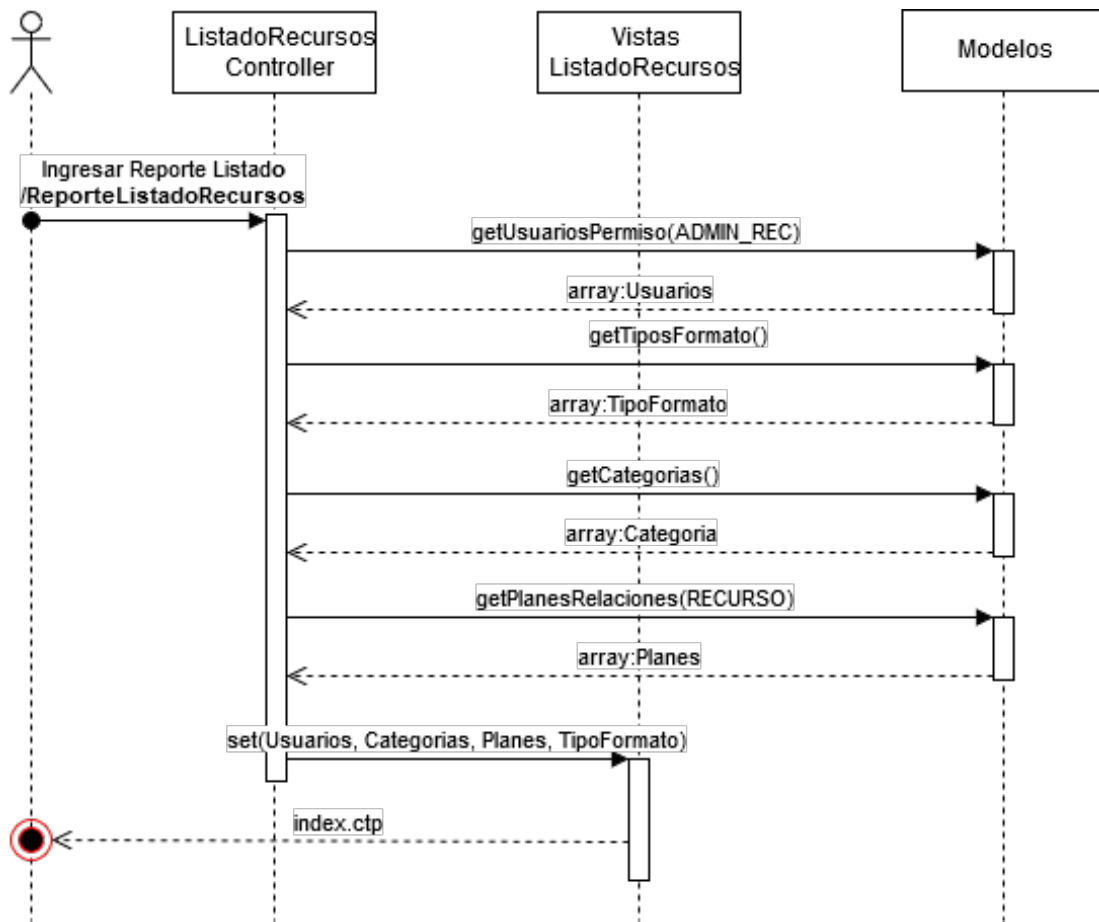


Figura 2.15: Diagrama de secuencia ingreso “Listado de recursos educativos”

En la figura 2.15 presentamos el diagrama de secuencia que presenta la interacción de los distintos componentes de la funcionalidad descrita.

2. Búsqueda de recursos educativos

- a) Esta funcionalidad inicia una vez que el usuario ha ingresado al módulo de Listado de recursos educativos.
- b) El usuario ingresa la información deseada en los criterios de búsqueda y da click en el botón de buscar.
- c) El sistema realiza la búsqueda en la base de datos con los criterios ingresados por el usuario.
- d) El sistema muestra los recursos educativos coincidentes con los criterios de búsqueda ingresados por el usuario.

En la figura 2.16 presentamos el diagrama de secuencia que presenta la interacción de los distintos componentes de la funcionalidad descrita.

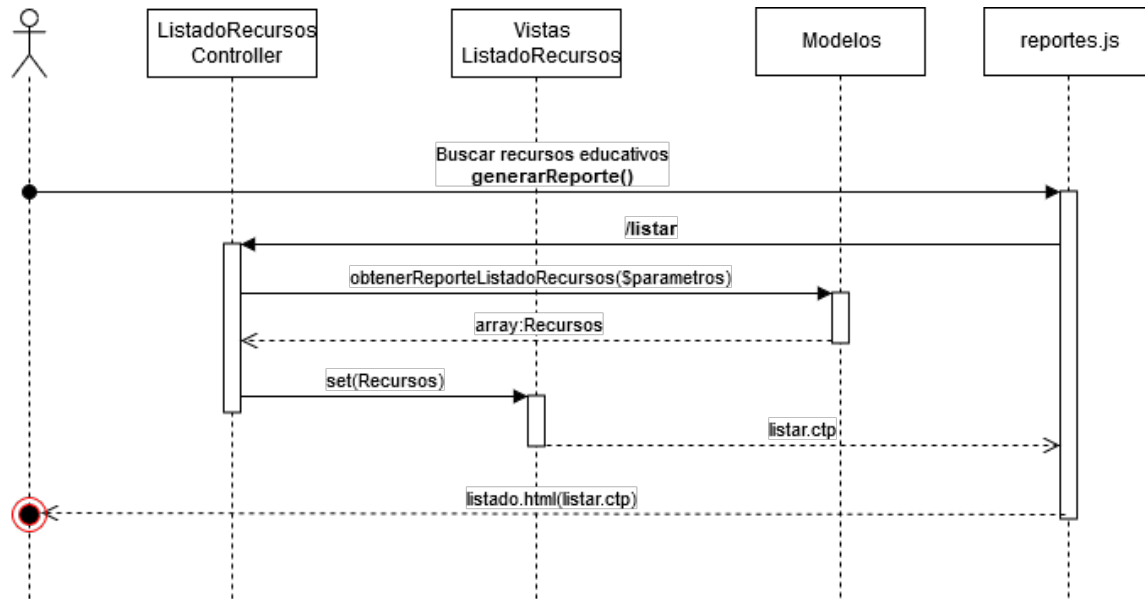


Figura 2.16: Diagrama de secuencia Listado de recursos educativos búsqueda de recursos

3. Exportar a Excel el listado de recursos educativos, a continuación describiremos la interacción del usuario con el sistema:

- a) Esta funcionalidad inicia después de presentarse el listado de recursos educativos, con los resultados coincidentes, con los criterios de búsqueda.
- b) El usuario selecciona el botón “Exportar a Excel”.
- c) El sistema realiza la búsqueda en la base de datos con los criterios ingresados previamente por el usuario.
- d) El sistema genera un archivo Excel con el listado de los recursos coincidentes.
- e) El sistema permite descargar el archivo Excel con el listado de recursos.

En la figura 2.17 presentamos el diagrama de secuencia que presenta la interacción de los distintos componentes de la funcionalidad descrita.

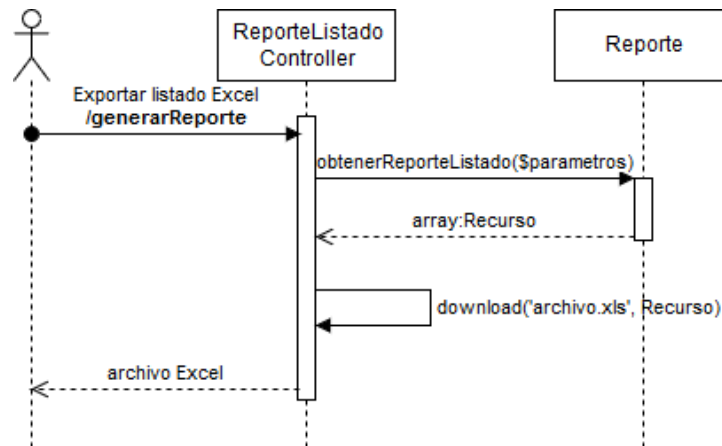


Figura 2.17: Diagrama de secuencia exportar listado de recursos educativos a Excel

2.8.2. Diagrama de secuencia del módulo: seguimiento de evaluaciones

En esta sección describiremos las funcionalidades antes mencionadas del módulo de seguimiento de evaluaciones de recursos educativos.

1. Ingreso “Evaluación de recursos”
 - a) Esta funcionalidad inicia cuando el usuario selecciona en el menú del sistema Evaluación de recursos en la sección de Elementos RUA.
 - b) El sistema carga los distintos catálogos necesarios para presentar los criterios de búsqueda.
 - c) El sistema muestra los criterios de búsqueda del reporte.
 - d) El sistema envía una petición para obtener las evaluaciones.
 - e) El sistema realiza la búsqueda en la base de datos de las evaluaciones registradas.
 - f) El sistema muestra las evaluaciones registradas.
2. Búsqueda de evaluaciones. Esta funcionalidad funciona como la parte anterior a partir del punto d) con excepción que el usuario ingresa los criterios de búsqueda. En la figura 2.18 presentamos el diagrama de secuencia que presenta la interacción de los distintos componentes de la funcionalidad descrita.

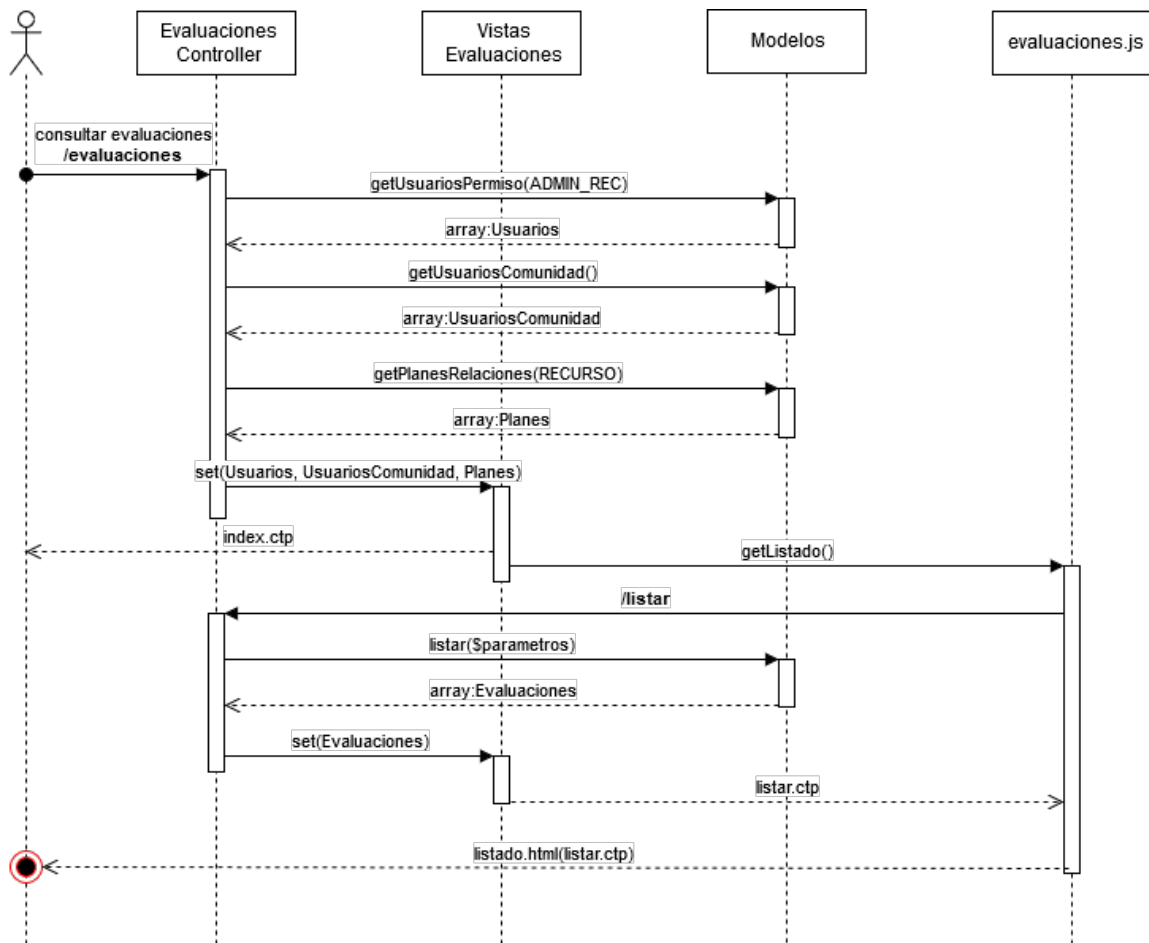


Figura 2.18: Diagrama de secuencia ingreso a “Evaluaciones de recursos educativos”

3. Exportar Listado/Rúbrica Excel

- a) Esta funcionalidad inicia después de presentarse el listado de las evaluaciones.
- b) El usuario selecciona el botón “Exportar Excel” el cual muestra un menú desplegable del cual el usuario puede seleccionar: “Listado” o “Evaluación”.
- c) El sistema realiza la búsqueda de las evaluaciones que coincidan con los criterios de búsqueda ingresados previamente por el usuario.
- d) El sistema genera un archivo Excel con las evaluaciones coincidentes.
- e) El sistema permite descargar el archivo Excel con las evaluaciones.

En la figura 2.19 presentamos el diagrama de secuencia que presenta la interacción de los distintos componentes de las funcionalidades descritas.

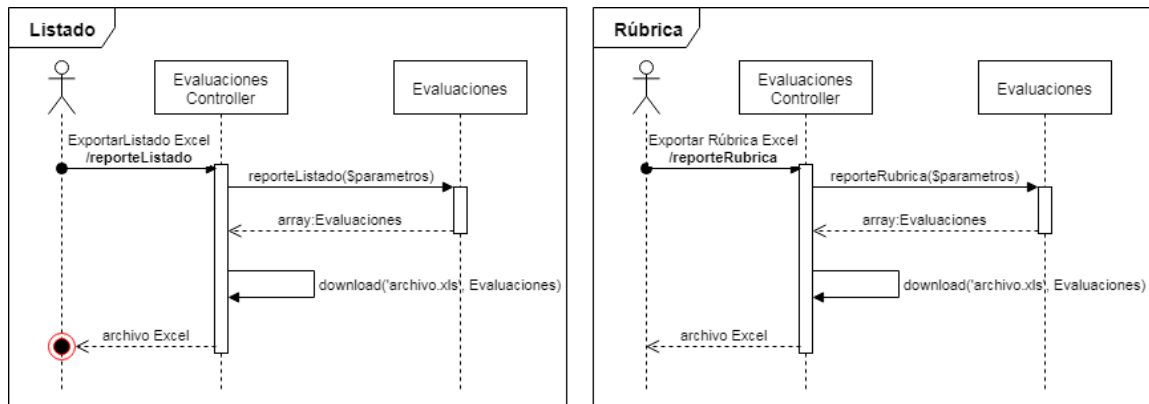


Figura 2.19: Diagrama de secuencia Exportación a excel Rúbrica y Listado

4. Consulta de evaluación

- a) Esta funcionalidad inicia después de presentarse el listado de las evaluaciones.
- b) El usuario selecciona el botón “Consultar evaluación” en algún renglón del listado de evaluaciones.
- c) El sistema busca en la base de datos la evaluación seleccionada por el usuario.
- d) El sistema muestra los resultados de las preguntas de las distintas secciones de la evaluación. Adicionalmente presenta dos campos: Modificar estado y Observaciones; donde el usuario puede ingresar datos.
- e) El usuario ingresa los datos en los campos que muestra el sistema y da click en guardar.
- f) El sistema guarda los datos ingresados por el usuario de la evaluación seleccionada.
- g) El sistema muestra el mensaje “Los cambios han sido guardados correctamente”
- h) El sistema redirecciona al usuario al listado de evaluaciones registradas.

En la figura 2.20 presentamos el diagrama de secuencia que presenta la interacción de los distintos componentes de la funcionalidad descrita.

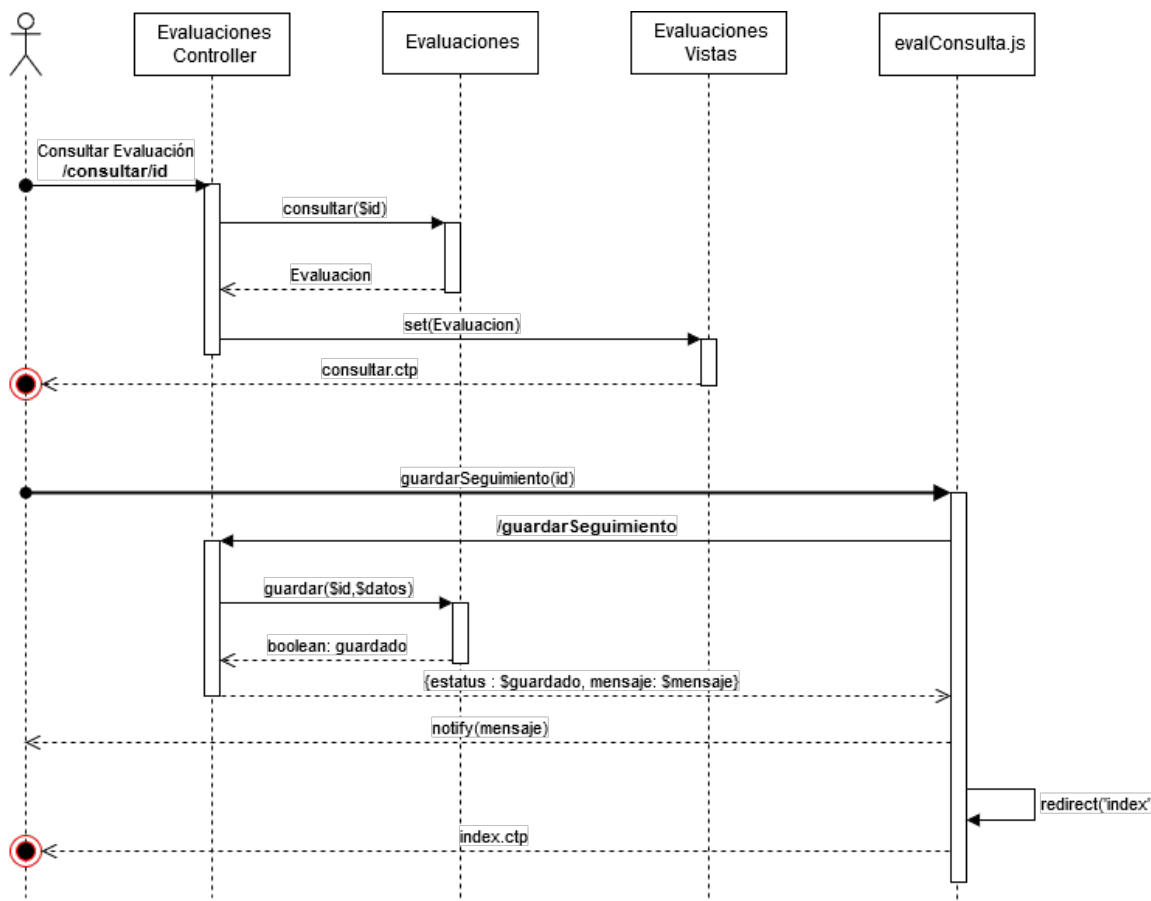


Figura 2.20: Diagrama de secuencia Consulta de evaluación

Resumen

En este capítulo describimos de manera general los módulos que componen la RUA IES, adicionalmente detallamos los requerimientos de los módulos en los cuales participé, a partir de los cuales detallamos el diseño de dos módulos de la RUA IES en los cuales participé. Describimos la estructura del proyecto y detallamos los componentes de los módulos: *Listado de recursos educativos* y *Seguimiento de evaluación de recursos educativos*. Con la presentación de la vista estática y vista dinámica de ambos módulos, podemos observar la similitud entre la interacción de sus componentes; estos módulos representan el funcionamiento general de cualquier módulo que forma parte de la administración del sistema, debido a que las diferencias entre estos es, principalmente, en los datos consultados, por lo cual en el siguiente capítulo, describiremos únicamente las funcionalidades de ingreso y

búsqueda; del módulo *Seguimiento de evaluaciones de recursos*.

Realizamos la descripción detallada del módulo de *Administración de asociaciones*, para entender la búsqueda en el árbol del programa de estudios, la cual detallaremos en el siguiente capítulo.

3. Implementación

En este capítulo presentaremos la implementación de las funcionalidades: *Ingreso Evaluaciones de recursos* y *Búsqueda de evaluaciones de recursos educativos*; del módulo *Seguimiento de evaluaciones de recursos educativos*. La implementación se hizo principalmente con base en el caso de uso correspondiente al módulo. Para entender mejor la implementación realizada detallaremos algunas características de CakePHP. Adicionalmente ejemplificaremos la *búsqueda* sobre el *programa de estudios* utilizada en la funcionalidad de *búsqueda de evaluaciones*.

3.1. Detalle de la implementación del módulo: Seguimiento de evaluaciones

Comenzaremos con describir los componentes que comprenden la implementación del módulo de seguimiento de evaluaciones. Detallaremos los estándares utilizados por el framework CakePHP para comprender la adaptación del patrón de diseño MVC brindada por el framework.

El módulo de Seguimiento de evaluaciones de recursos educativos se compone de tres componentes principales siguiendo el patrón de diseño MVC.

3.1.1. Modelo

El archivo **Evaluacion.php** corresponde al Modelo del módulo. Los modelos dentro de CakePHP representan una tabla de nuestra base de datos. Todos los modelos dentro de CakePHP deben heredar de la clase **Model**, la cual es provista por el núcleo del framework. En la figura 3.1 presentamos la declaración de la clase **Evaluacion** contenida en el archivo **Evaluacion.php**.

Los modelos deben de declarar el nombre de la tabla usada (línea 13 Figura 3.1) y la llave primaria (línea 14 Figura 3.1) de esta. Adicionalmente, podemos utilizar la secuencia

(línea 16 Figura 3.1) utilizada para la asignación del identificador y el orden en el cual se buscarán la información de la tabla (línea 14 Figura 3.1).

```
1 <?php
2
3 App::uses('RuaAppModel', 'Model');
4
5 /**
6  * CakePHP Evaluacion.
7  *
8  * @author BLS Jonathan Blasio
9  */
10 class Evaluacion extends RuaAppModel {
11
12     public $name = 'Evaluacion';
13     public $useTable = 'evaluacion';
14     public $primaryKey = 'evaluacion_id';
15     public $order = 'evaluacion_id';
16     public $sequence = 'seq_evaluacion';
```

Figura 3.1: Evaluacion.php

3.1.2. Controlador

El archivo **EvaluacionesController.php** corresponde al controlador del módulo. La principal actividad de los controladores dentro de CakePHP, es manejar la información enviada por el usuario, para que sea procesada por las clases necesarias, para generar una respuesta a la petición realizada. Todos los controladores dentro de CakePHP deben heredar de la clase **AppController**, la cual es provista por el núcleo del framework. En la figura 3.2 presentamos la declaración de la clase **EvaluacionesController** contenida en el archivo **EvaluacionesController.php**.

Los Controladores deben declarar los modelos (línea 14 Figura 3.2) utilizados por el controlador; adicionalmente se declaran los componentes utilizados (línea 12 Figura 3.2) y las clases que serán utilizadas en las vistas (línea 13 Figura 3.2).

Los métodos declarados en los controladores deben tener una vista la cual debe localizarse en la carpeta `App/Views/NombreControlador`. Los métodos dentro del controlador de evaluaciones son: *index*, *listar*, *reporteListado*, *reporteRubrica*, *consulta* y *guardarSeguimiento*. Al finalizar la ejecución de cada método del controlador, CakePHP cargará de manera automáticamente la vista asociada a este.

```

2 App::uses('AppController', 'Controller');
3 App::uses('Email', 'Email');
4
5 /**
6  * Controlador de Evaluaciones.
7  *
8  * @author BLS Jonathan Blasio
9  */
10 class EvaluacionesController extends AppController {
11
12     public $components = array('RequestHandler', 'Paginator');
13     public $helpers = array('Js');
14     public $uses = array('Evaluacion', 'GenerarReporte');

```

Figura 3.2: EvaluacionesController.php

3.1.3. Vistas

La carpeta **Evaluaciones** contiene las vistas correspondientes a los métodos del controlador `EvaluacionesController`; las vistas del módulo de seguimiento de evaluaciones están contenidas en los archivos: *index.ctp*, *listar.ctp* y *consulta.ctp*. CakePHP automáticamente liga los nombres de los métodos del controlador con sus respectivas vistas. Las vistas del framework tiene asociadas clases que nos permiten trabajar con etiquetas de HTML. Adicionalmente a los elementos de HTML de las vistas, tenemos archivos de *javascript* utilizados dentro de estas, para mejorar la interacción del usuario con el módulo. Los scripts utilizados en el módulo se encuentran en la carpeta de **webroot** y son: *evaluaciones.js* y *evalConsulta.js*.

3.2. Implementación de la funcionalidad: Ingreso seguimiento de evaluaciones

Como mencionamos en el capítulo anterior en la descripción de la funcionalidad de *Ingreso de evaluaciones de recursos*; esta funcionalidad comienza cuando el usuario selecciona en el menú del sistema la opción “**Evaluación recursos**”, esta opción está implementada como un hipervínculo, el cual dirige al usuario a la URL `/Evaluaciones`, esta URL es mapeada por CakePHP al controlador **EvaluacionesController** a su método `index`. En la figura 3.3 presentamos el código del método `index` del controlador **EvaluacionesController**. El método `index` se encarga únicamente de obtener los catálogos (líneas 23-34 Figura 3.3) de la base de datos, necesarios para presentar los criterios de búsqueda. Después se verifica si existen evaluaciones registradas (líneas 3.7 Figura 3.3) y por último se cargan los datos

obtenidos en la vista.

```
22     public function index() {
23         $usuariosConsulta =
24             $this->Evaluacion->Usuario->getUsuariosPorPermisos(
                ADMIN_RECURSOS);
25         $usuarios = array();
26         foreach ($usuariosConsulta as $usuario) {
27             $usuarios[] = array(
28                 'id' => $usuario['Usuario']['usuario_id'],
29                 'value' => $usuario['Usuario']['nombre_completo'],
30                 'label' => $usuario['Usuario']['nombre_completo']
31             );
32         }
33         $usuariosComunidad = $this->__getUsuariosComunidad();
34         $planesEstudio = $this->Evaluacion->Elemento->NodoPrimerNivel
                ->getPlanesConRelaciones(RECURSO);
35
36         $numeroEvaluaciones = $this->Evaluacion->find('count');
37
38         $this->set(compact('usuarios', 'usuariosComunidad', '
                planesEstudio', 'numeroEvaluaciones'));
39     }
```

Figura 3.3: Método index EvaluacionesController

La vista carga los archivos de *javascript* y hojas de estilo necesarias (línea 2-6 Figura 3.4), inicializa las variables de *javascript* necesarias en esta vista para cumplir con los requerimientos (línea 9-15 Figura 3.4). Estas variables incluyen URL asociadas a acciones correspondientes al módulo y variables para la presentación de los criterios de búsqueda. En la parte visible por el usuario, se cargan los criterios de búsqueda (línea 3.11-44 Figura 3.4); estos criterios se encuentran en una vista distinta y es cargada con el uso de la función **element**. En la última sección de la vista se agrega un contenedor para cargar el listado de evaluaciones (línea 46 Figura 3.4) que será presentada al usuario; este listado con las evaluaciones es cargada a través de *javascript* de manera asíncrona.

Después de que el usuario recibe la vista del index del módulo, se carga el archivo de *javascript* **evaluaciones.js**. Este archivo se encarga de cargar el listado de evaluaciones y realizar la búsqueda de evaluaciones a través de los criterios ingresados.

Aunque la carga del listado se realiza en el index de manera asíncrona, omitiremos la descripción en esta sección ya que coincide con la búsqueda de evaluaciones que describiremos a continuación.

```

2 echo $this->Html->script("evaluaciones");
3 echo $this->Html->script(RUA_JS . "jquery-ui.min");
4 echo $this->Html->script(RUA_JS . "tableHeadFixer");
5 echo $this->Html->css(RUA_CSS . "jquery-ui.min");
6 echo $this->Html->css(RUA_CSS . "jquery-ui.structure.min");
7 ?>
8 <script>
9     var urlListado = '<?php echo $this->Html->url('listar')?>';
10    var urlReporteListado = '<?php echo $this->Html->url('
11        reporteListado')?>';
12    var urlReporteRubrica = '<?php echo $this->Html->url('
13        reporteRubrica')?>';
14
15    var esSubmit = false;
16    var usuarios = <?php echo json_encode($usuarios);?>;
17    var usuariosComunidad = <?php echo json_encode($usuariosComunidad)
18        ;?>;
19 </script>
20
21 <hr/>
22 <div class="row">
23     <div class="col-xs-12" id="busqueda" style="display:none">
24         <?php echo $this->element('evaluaciones/busqueda', array('
25             planEstudios' => $planesEstudio)); ?>
26     </div>
27 </div>
28
29 <div id="listadoEval">

```

Figura 3.4: Vista index.ctp

En la figura 3.5 presentamos la vista del módulo que se presenta al usuario.

Evaluación de recursos

Criterios de búsqueda

Seleccione los criterios para consultar las evaluaciones.

Título del recurso

Id del recurso

Usuario que evaluó el recurso

Último usuario que dio seguimiento a la evaluación

Estado de seguimiento

Plan/Elemento del programa de estudios

Calificación global

Dictamen

Fecha de evaluación

Desde

Hasta

Figura 3.5: Criterios de búsqueda evaluaciones

3.3. Implementación de la funcionalidad: Búsqueda de evaluaciones de recursos educativos

La funcionalidad de búsqueda de evaluaciones se encuentra contenida en el index de este módulo; esta búsqueda se realiza con ayuda del script **evaluaciones.js**, el cual se encarga de procesar los datos ingresados en los criterios de búsqueda y enviarlos de manera asíncrona para presentar el listado con las evaluaciones buscadas.

La búsqueda petición se realiza una vez que el usuario ha presionado el **botón "Buscar"** en el formulario presentado en el index, después de ingresar los criterios de búsqueda deseados. En la figura 3.6 se presenta la función con la cual se envía la petición al controlador para realizar la búsqueda de evaluaciones de recursos educativos.

La búsqueda de evaluaciones de recursos se realiza a través de una petición *ajax* (línea 2-10 Figura 3.6) a la URL "**Evaluaciones/listar**" (línea 3 Figura 3.6). Dentro de la petición se envían los datos ingresados por el usuario por medio del formulario (línea 4 Figura

3.6). Finalmente una vez que la función regresa el resultado de la petición, los datos son cargados (línea 8 Figura 3.6) en el elemento de HTML `#listadoEval` (línea 46 Figura 3.4) que se encuentra en el index del módulo, previamente cargado.

```

1 function listar(form) {
2     $.ajax({
3         url: urlListado,
4         data: form.serialize(),
5         type: 'get',
6         dataType: 'html',
7         success: function(data, status){
8             $('#listadoEval').html(data);
9         }
10    });
11 }

```

Figura 3.6: Función de *javascript* listar `evaluaciones.js`

El método `listar` del controlador `EvaluacionesController` se encarga de procesar la petición realizada por el archivo `evaluaciones.js`. En la figura 3.7 presentamos el código del método `listar` del controlador `EvaluacionesController`.

```

48 public function listar() {
49     $condicionesBase = $this->Evaluacion->configPaginateAdmin['conditions'];
50
51     if (!empty($this->request->query)) {
52         $this->Evaluacion->filtrar($this->request->query);
53     }
54
55     $hayFiltros = !($condicionesBase == $this->Evaluacion->
56         configPaginateAdmin['conditions']);
57     $listadoNuevo = MENSAJE_SIN_REGISTRO;
58     $sinResultados = MENSAJE_SIN_COINCIDENCIAS_ELEMENTOS;
59     $mensajeVacio = ($hayFiltros) ? $sinResultados : $listadoNuevo;
60
61     $this->Paginator->settings = $this->Evaluacion->configPaginateAdmin;
62     $evaluaciones = $this->Paginator->paginate();
63
64     $this->set(compact('evaluaciones', 'mensajeVacio'));
65     $this->render('listar', 'ajax');
66 }

```

Figura 3.7: Método listar `EvaluacionesController`

El método `listar` recibe los datos del usuario por medio del formulario; en CakePHP estos datos son almacenados en la variable del controlador `$this → request → query`. Posteriormente los agrega como parámetros a la función `filtrar` (línea 50-52 Figura 3.7) del modelo que se encarga de buscar las evaluaciones en la base de datos. Después se verifica que la

consulta tenga resultados. En caso de no tener resultados, si el usuario no ingresó ningún criterio de búsqueda se presenta el mensaje sin registro de de evaluaciones (línea 56 Figura 3.7); en caso contrario presenta el mensaje sin coincidencias en la búsqueda (línea 57 Figura 3.7). Con los criterios de búsqueda creados de acuerdo a los datos ingresados por el usuario el controlador utiliza el componente **Paginate**, proporcionado por CakePHP, que permite paginar los resultados de la búsqueda. Por último se envían las evaluaciones y mensaje a la vista.

En la figura 3.8 se presenta el listado de evaluaciones mostrada al usuario, en el listado se muestran los datos de las evaluaciones descritos en el caso de uso, los campos del listado pueden ser ordenados como lo indica el caso de uso, adicionalmente se muestra el total de registros coincidentes con la búsqueda y los datos de la página que se está mostrando. Podemos observar que se presentan las funcionalidades de “Exportar Excel Listado” y “Exportar Excel Rúbrica” en el listado. Adicionalmente cada renglón del listado contiene un botón para ingresar a la consulta de la evaluación seleccionada.

La búsqueda de evaluaciones en la base de datos será descrita en la siguiente sección mostrando la consulta utilizada y describiendo la búsqueda sobre el programa de estudios.

Exportar Excel

Listado Evaluaciones 24470 registros

Acciones	Id	Id del recurso	Título	Observaciones	Plan/Elemento del programa de estudios	Fecha de evaluación	Usuario evaluó recurso
	24706	9470	Álgebra		<ul style="list-style-type: none"> /Colegio de Ciencias y Humanidades -- Plan 2003/Primer semestre/Matemáticas I (Álgebra y Geometría) /Colegio de Ciencias y Humanidades/Primer semestre/Matemáticas I 	22/10/2018	Jonathan B Leonardo
	24705	19996	Marcadores textuales	El recurso es útil y da apoyo al aprendizaje.	<ul style="list-style-type: none"> /Colegio de Ciencias y Humanidades/Tercer semestre/Taller de Lectura, Redacción e Iniciación a la Investigación Documental III/Unidad I. Texto dramático. Representación teatral. Reseña crítica/Marcadores textuales. /Colegio de Ciencias y Humanidades/Tercer semestre/Taller de Lectura, Redacción e Iniciación a la Investigación Documental III/Unidad II. Editorial y caricatura política. Comentario analítico/Marcadores textuales. /Escuela Nacional Preparatoria/Cuarto año/1402 Lengua Española/Unidad 5. Leer e interpretar: lo icónico verbal/5.7 Fases del proceso de producción de un texto icónico verbal relacionado con circunstancias actuales (sentimiento de vacío, soledad, superficialidad de la vida, crítica social o consumismo) o de los poemas de vanguardia leídos. /d) Textualización: uso de marcas lingüísticas como signos de puntuación, conectores textuales. 	21/09/2017	MARGARIT, BERENICE MUÑOZ LO

Mostrando del 1 al 20 de 24470 registros

Figura 3.8: Listado de evaluaciones

3.4. Búsqueda de evaluaciones de recursos educativos

La búsqueda de evaluaciones de recursos educativos es realizada en el Modelo **Evaluacion**, la consulta es creada con ayuda del *QueryBuilder* proporcionado por CakePHP; En la Tabla 3.1 presentamos los campos correspondientes a cada tabla del listado.

Tabla	Campos	Campo en el listado
evaluacion	evaluacion_id	Id
	elemento_id	Elemento Id
	eval_observaciones	Observaciones
	eval_fecha_evaluacion	Fecha de evaluación
	eval_dictamen	Dictamen
	eval_calificacion	Calificación
	eval_estatus_atencion	Estado de seguimiento
elemento	elem_nombre	Título
recurso	recur_url	Fuente del recurso
usuario	usu_nombre	Último usuario que dio seguimiento a la evaluación
	usu_primer_apellido	
	usu_segundo_apellido	
usuario_comunidad	usu_com_nombre	Usuario que evaluó el recurso
	usu_com_primer_apellido	
	usu_com_segundo_apellido	

Tabla 3.1: Tabla de campos en el listado

Adicionalmente a las tablas usadas en la consulta es usada la función **get_relaciones_ascendencia** la cual regresa una tabla, con la cual obtenemos las rutas (línea 11, 18-20 Figura 3.9) donde se encuentra clasificado el recurso evaluado; para la presentación del campo en el listado “*Plan/Elemento del programa de estudios*”. En la siguiente sección describiremos una función análoga a esta función, para entender cómo se expande el árbol del programa de estudios.

En la figura 3.9 presentamos la consulta generada con el QueryBuilder.


```

1 SELECT
2     EV.evaluacion_id, EV.elemento_id, EL.elem_nombre,
3     EL.elemento_id, R.recur_tipo_recurso, R.recur_url,
4     EV.eval_observaciones, EV.eval_fecha_evaluacion,
5     EV.eval_dictamen, EV.eval_calificacion,
6     EV.eval_estatus_atencion,
7     concat(U.usu_nombre, ' ', U.usu_primer_apellido,
8            ' ', U.usu_segundo_apellido),
9     concat(UC.usu_com_nombre, ' ', UC.usu_com_primer_apellido,
10            ' ', UC.usu_com_segundo_apellido),
11     array_agg(GRA.ruta)
12 FROM   evaluacion EV
13 JOIN   elemento EL USING (elemento_id)
14 JOIN   recurso R USING (elemento_id)
15 JOIN   usuario U ON U.usuario_id = EV.usuario_id
16 JOIN   usuario_comunidad UC
17         ON EV.usuario_comunidad_id = UC.usuario_comunidad_id
18 LEFT JOIN get_relaciones_ascendencia(EV.elemento_id) GRA
19         ON GRA.elemento_id_destino = EV.elemento_id
20         AND GRA.tipo_elemento_id_origen = 4
21 GROUP BY
22     EV.evaluacion_id, EV.elemento_id, EL.elem_nombre,
23     EL.elemento_id, R.recur_tipo_recurso, R.recur_url,
24     EV.eval_observaciones, EV.eval_fecha_evaluacion,
25     EV.eval_dictamen, EV.eval_calificacion,
26     EV.eval_estatus_atencion, U.usu_nombre,
27     U.usu_primer_apellido, U.usu_segundo_apellido,
28     UC.usu_com_nombre, UC.usu_com_primer_apellido,
29     UC.usu_com_segundo_apellido
30 ;

```

Figura 3.9: Consulta evaluaciones

3.4.1. Implementación función `get_relaciones_descendencia`

La función `get_relaciones_descendencia` se encarga de expandir el árbol del programa de estudios, esta función recibe como parámetro **padre**, que corresponde al elemento a partir del cual se expandirá el árbol del programa de estudios. La función obtendrá todas las relaciones que existen a partir del elemento seleccionado. La implementación de esta función se realizó con el uso de las *consultas recursivas* descritas en el Capítulo 1. Recordemos que las funciones recursivas inician con una consulta no recursiva, en la figura 3.10 mostramos la parte no recursiva de la consulta.

```

38 WITH RECURSIVE relaciones(
39     elemento_id_origen, elemento_id, tipo_elemento_id_origen,
40     tipo_elemento_id_destino, tipo_relacion_id,
41     ruta, camino, ciclo, ordenacion)
42 AS(
43     SELECT
44         E.elemento_id, E.elemento_id, E.tipo_elemento_id,
45         E.tipo_elemento_id, 4, '', ARRAY[E.elemento_id],
46         false, ARRAY[1]
47     FROM     elemento E
48     WHERE    E.elemento_id = padre
49 UNION

```

Figura 3.10: Función `get_relaciones_descendencia` parte no recursiva

La parte no recursiva de la función inicia con los datos del elemento **padre**, que será el elemento a partir del cual expandiremos el árbol del programa de estudios. Los datos que obtendremos de esta función son:

- **elemento_id_origen**: Corresponde al `elemento_id` de inicio y lo inicializamos con el id del elemento **padre**, este dato no cambiará durante la consulta.
- **elemento_id**: Corresponde al `elemento_id` con el cual se realiza está relacionado el elemento, estos id los inicializamos con el id del elemento **padre**, para indicar que se encuentra relacionado con sí mismo.
- **tipo_elemento_id_origen**: Corresponde al id del tipo de elemento origen de la relación. Lo inicializamos con el tipo de elemento del elemento **padre**.
- **tipo_elemento_id_destino**: Corresponde al id del tipo de elemento destino de la relación. Lo inicializamos con el tipo de elemento del elemento **padre**.
- **tipo_relacion_id**: Corresponde al tipo de relación con la cual se asocian los elementos. Lo inicializamos con el tipo de relación que corresponde a una relación de **descendencia**.
- **ruta**: Corresponde a una cadena que contiene la ruta tomada para llegar al elemento. Lo inicializamos con una cadena vacía.
- **camino**: Corresponde a un arreglo que contiene los id's de los elementos que han sido recorridos para llegar al elemento.
- **ordenación**: Corresponde al orden de la relación del elemento. Esto nos permite ordenar los elementos para mostrarlos con un recorrido a profundidad (DFS). Lo inicializamos con un arreglo que contiene un 1.

- **ciclo:** Corresponde a un booleano que nos indica si la consulta ha entrado en un ciclo. Lo inicializamos en **false**.

En la figura 3.11 presentamos la parte recursiva de la consulta, en esta consulta podemos hacer referencia a los datos de la tabla **relaciones** (línea 38-42 Figura 3.10), que fueron generados en la iteración anterior.

En el paso recursivo de la consulta utilizamos la tabla **relacion** para realizar la expansión del árbol del programa de estudios, esto lo hacemos con ayuda de los datos anteriores para expandir a partir del elemento hasta el que hemos explorado. A continuación describiremos cómo se llenan los campos en la consulta recursiva (línea 50-61 Figura 3.11).

- **elemento_id_origen:** Este dato no varía en las iteraciones y corresponde al **elemento_id** elemento a partir del cual se inició(**padre**) la expansión del árbol del programa de estudios.
- **elemento_id:** Corresponde al **elemento_id** con el cual se realiza la relación, este valor lo obtenemos de la tabla **relacion** del campo **elemento_id_destino**.
- **tipo_elemento_id_origen:** Corresponde al tipo de elemento correspondiente a la tabla **relacion** del campo **elemento_id_origen**.
- **tipo_elemento_id_destino** Corresponde al tipo de elemento correspondiente a la tabla **relacion** del campo **elemento_id_destino**.
- **tipo_relacion_id:** Corresponde al tipo de relación con la que se asocian los elementos.
- **ruta:** En el paso recursivo agregamos a la cadena el campo **elem_nombre** del elemento con el **elemento_id** igual al de la tabla **relacion** del campo **elemento_id_origen**.
- **camino:** En el paso recursivo agregamos el id del elemento correspondiente a la tabla **relacion** del campo **elemento_id_destino**.
- **ordenacion:** En el paso recursivo agregamos el campo **rel_orden_horizontal**, esto nos permitirá ordenar los datos para presentarlos como una búsqueda a profundidad (DFS).
- **ciclo:** En el paso recursivo verificamos si el elemento que visitamos no lo hemos expandido con anterioridad. Esto lo hacemos verificando que **elemento_id_destino** no se encuentre en el campo **camino** que se ha generado hasta el momento.

```

49 UNION
50 SELECT
51     rs.elemento_id_origen, e.elemento_id,
52     rs.tipo_elemento_id_destino, e.tipo_elemento_id,
53     r.tipo_relacion_id,
54     CASE WHEN e2.tipo_elemento_id IN (3,4)
55         THEN rs.ruta || '/' || e2.elem_nombre
56         ELSE rs.ruta
57     END,
58     rs.camino || e.elemento_id,
59     e.elemento_id = ANY(rs.camino),
60     rs.ordenacion || r.rel_orden_horizontal
61     END
62 FROM     relacion r
63 JOIN relaciones rs ON rs.elemento_id = r.elemento_id_origen
64 JOIN elemento e ON r.elemento_id_destino = e.elemento_id
65 JOIN elemento e2 ON rs.elemento_id = e2.elemento_id
66 WHERE
67     NOT rs.ciclo AND e.elem_estatus_publicacion != status
68     AND e2.elem_estatus_publicacion != status AND
69     ((r.tipo_relacion_id = 4 AND rs.tipo_relacion_id = 4 AND
70      (e.tipo_elemento_id = 3 OR e.tipo_elemento_id = 4)
71      AND (e2.tipo_elemento_id = 3 OR e2.tipo_elemento_id = 4))
72     OR
73     (r.tipo_relacion_id != 4 AND rs.tipo_relacion_id = 4 AND
74      (e2.tipo_elemento_id = 3 OR e2.tipo_elemento_id = 4))
75     )
76 )
77 SELECT
78     elemento_id_origen, elemento_id,
79     tipo_elemento_id_origen, tipo_elemento_id_destino,
80     tipo_relacion_id, ruta, camino,
81     relacion_id_orden, ordenacion
82 FROM relaciones RS
83 ;

```

Figura 3.11: Función get_relaciones_descendencia parte recursiva

Por último describiremos las reglas para expandir el programa de estudios, las cuales se encuentran implementadas como condiciones en la consulta recursiva. Las reglas de asociación fueron descritas en el Capítulo 2 de este documento.

Primero verificamos que no se haya entrado en un ciclo con ayuda del campo **ciclo** (línea 67 Figura 3.11). Después verificamos que las relaciones de los elementos sean de tipo *descendencia*(4) y que los elemento sean de tipo *nodo primer nivel*(4) o *nodo*(3) [línea 69-71 Figura 3.11]. O verificamos que la asociación hasta el paso anterior sea de tipo *descen-*

dencia(4); y la relación actual no sea de tipo *descendencia* y el elemento desde el cual se realiza la asociación (*relacion.elemento_id.origen*) sea de *tipo nodo primer nivel*(4) o *nodo*(3) [Línea 73-76 Figura 3.11]. Con estas condiciones garantizamos que expandimos solo los elementos que corresponden al árbol del programa de estudios, el cual es formado únicamente por *nodos de primer nivel* y *nodos* con tipo de relación *descendencia*. Ya que pueden existir elementos de tipo *nodo de primer nivel* y *nodo*; que se encuentran asociados por una relación que **no** es de *descendencia* sin embargo estos nodos no corresponden al programa de estudios y no deben de ser expandidos.

Esta función nos permite encontrar todos los elementos que componen el árbol del programa de estudio, lo cual nos permite realizar la búsqueda de recursos educativos a partir de un nodo seleccionado por el usuario como fue descrita en la sección 2.5.2 Búsqueda en el programa de estudios de recursos educativos.

Resumen

A lo largo de este capítulo describimos la implementación de las funcionalidades de ingreso y búsqueda, del módulo Seguimiento de evaluaciones de recursos. Esta descripción nos permitió profundizar en el funcionamiento de CakePHP, detallando la implementación del patrón de diseño MVC realizada por el framework. Adicionalmente detallamos la búsqueda en el árbol del programa de estudios; cuya función es utilizada en 80 % de los módulos de administración y forma parte fundamental de la consulta de recursos en el portal de RUA IES, con esto logrando uno de los objetivos principales de la RUA IES; la cual corresponde a la búsqueda de recursos educativos en el árbol del programa de estudios.

4. Conclusiones

4.1. Resumen general

A lo largo de este reporte hemos descrito el proyecto RUA IES, los objetivos generales del proyecto, su propósito y ventajas, también se describió la forma de trabajo del equipo involucrado y se detallaron las tecnologías utilizadas en su desarrollo. Esto permitió describir los requerimientos principales del sistema, y la división en sus diferentes módulos. Finalmente detallamos el diseño de dos módulos, para entender de mejor manera, su implementación: Seguimiento de evaluaciones de recursos educativos y Listado de recursos educativos.

La búsqueda en el árbol del programa de estudios, en el módulo de Seguimiento de evaluaciones, representa solo un campo de búsqueda dentro del sistema, sin embargo esta búsqueda se encuentra en distintos módulos del sistema, haciendo de esta funcionalidad una parte fundamental de la RUA IES. Por esta razón durante los tres capítulos se enfatizó esta característica de la RUA IES, partimos desde su definición hasta la implementación hecha en la base de datos para ser integrada en los distintos módulos que se necesite. Esta característica de la RUA IES se encuentra en todos los módulos que tienen relación con los elementos de la RUA IES (recursos educativos, planes de estudios y elementos del programa de estudios). Además es parte fundamental del módulo del portal.

La clasificación de los elementos de RUA IES en el plan de estudios, permite a los administradores del sistema enfocarse en los temas de su especialidad facilitando la búsqueda de los recursos educativos que sean de su interés, permitiendo una mejor clasificación de estos.

En el capítulo de diseño podemos observar una similitud muy clara entre los componentes de ambos módulos descritos, esto es una constante en el desarrollo de la RUA IES, que brinda uniformidad en el desarrollo, que permitirá un mantenimiento y evolución del sistema del sistema.

4.2. Experiencia en la metodología de trabajo

La constancia de los miembros del equipo de desarrollo durante la implementación de las tres versiones de la RUA IES, fortaleció el proceso de software seguido en el proyecto; esto permitió un conocimiento general del sistema por parte del equipo, lo cual ayudó a que cada miembro del equipo pudiera entender el código desarrollado por otro desarrollador, con esto se logró distribuir la carga del trabajo de desarrollo; siendo capaz cada desarrollador de cerrar incidencias de otros módulos cuando el desarrollador de dicho módulo estuviera saturado.

Además fue fundamental, para la correcta implementación del sistema, la comunicación entre los equipos de desarrollo y análisis, que a medida que avanzó el proyecto logró disminuir el número de incidencias reportadas por el equipo de pruebas, de 527 incidencias reportadas en la primera versión a 474 incidencias reportadas en la tercera, contando con 25 y 15 incidencias críticas en la primera y tercera versión respectivamente; adicionalmente se realizó de manera paralela el diseño y desarrollo del sistema en las últimas fases, reduciendo así el tiempo de desarrollo. En los ciclos de prueba realizados al sistema, se garantizó un desarrollo que cumplía cabalmente con la documentación del sistema; específicamente con los casos de uso de cada módulo, los cuales eran las bases para realizar el desarrollo del software y el plan de pruebas del sistema. Adicionalmente, hubo rotaciones en los miembros del equipo de pruebas, en distintas etapas del proyecto, lo cual brindó una nueva perspectiva del proyecto, logrando identificar errores que no se habían encontrado en etapas pasadas, y con esto se mejoró la confiabilidad de la RUA IES.

Además fue fundamental, para la correcta implementación del sistema, la comunicación entre los equipos de desarrollo y análisis, que a medida que avanzó el proyecto logró disminuir el número de incidencias reportadas por el equipo de pruebas; y realizar de manera paralela el diseño y desarrollo del sistema en las últimas fases, reduciendo así el tiempo de desarrollo. En los ciclos de prueba realizados al sistema, se garantizó un desarrollo que cumplía cabalmente con la documentación del sistema. Adicionalmente, hubo rotaciones en los miembros del equipo de pruebas, en distintas etapas del proyecto lo cual brindó una nueva perspectiva del proyecto, logrando identificar errores que no se habían encontrado en etapas pasadas, y con esto se mejoró la confiabilidad de la RUA IES.

4.3. Comentarios sobre el uso de tecnologías

CakePHP me permitió conocer el patrón de diseño MVC, lo cual me dio una perspectiva general del desarrollo web, gracias a la rigidez propia del framework logré adaptarme a frameworks más flexibles de forma más rápida. Una de las principales ventajas de CakePHP

en el proyecto fue la capa del modelo, ya que el QueryBuilder de CakePHP nos da una gran expresividad en el lenguaje SQL, brindando independencia del sistema manejador de base de datos usado. El QueryBuilder proporcionado por CakePHP nos permitió realizar todas las consultas del sistema haciendo uso de la paginación de datos, sin preocuparnos de esta.

La versión de CakePHP utilizada para el desarrollo de este proyecto era una de las últimas versiones del framework, lo que nos brindaba un marco de trabajo estable para comenzar el desarrollo; sin embargo actualmente con la evolución de PHP, esta versión de CakePHP no corresponde a la más reciente. En este sentido, una de las principales desventajas del uso de CakePHP es que no brinda un mecanismo de migración para actualizar la versión del framework.

4.4. Relación con mi carrera

Durante mi participación en este proyecto logré consolidar mis conocimientos de bases de datos, desde su diseño hasta el manejo de los datos. Adicionalmente, mi trabajo en el equipo de desarrollo me permitió desarrollar mis habilidades en el desarrollo de aplicaciones web con el uso de las tecnologías actuales. Con las bases de programación obtenidas a lo largo de la carrera, logré adaptarme al uso e integración de los distintos lenguajes de programación, frameworks y bibliotecas utilizadas para el desarrollo de la RUA IES; en ciertos casos sin tener conocimientos previos de las mismas.

4.5. Trabajo a futuro

Como mencionamos en el primer capítulo, la RUA permite ser adaptada a distintas IES; esto dio origen a un proyecto llamado RUA MX, el cual pretende concentrar todos los recursos educativos de las distintas instancias de la RUA IES y tenerlos clasificados de acuerdo con los planes de estudios de cada IES. Este proyecto fue desarrollado de manera paralela con la segunda versión de la RUA IES. Con el uso de CakePHP, gracias a los estándares impuestos por el framework, logramos un desarrollo consistente, esto a pesar de una curva de aprendizaje mayor para cumplir con el desarrollo de un módulo. Gracias a los estándares brindados por el framework y la continuidad de trabajo entre el equipo de desarrollo. Con el dominio del framework adquirido durante el desarrollo del proyecto se agilizó el desarrollo de los nuevos módulos de la versión dos de la RUA IES en conjunto con el desarrollo del proyecto de RUA MX.

Cuando dio inicio el proyecto RUA UNAM se encontraban clasificados 14802 recursos educativos a nivel bachillerato, divididos en dos planes de estudios; y 3609 recursos edu-

cativos nivel licenciatura, divididos en 29 planes de estudios distintos. Podemos observar que en la primera parte del proyecto de RUA IES, se logró generar una cantidad sustancial de recursos educativos para el nivel de bachillerato. Sin embargo esto deja el gran reto de seguir clasificando los recursos educativos generados por la universidad a nivel licenciatura y continuar el camino con las distintas especialidades brindadas por la UNAM.

Adicionalmente el proyecto de RUA UNAM, necesita que la RUA sea adoptada por distintas IES, para aumentar la oferta de recursos educativos de calidad para la comunidad.

A. Casos de uso

A.1. Caso de uso Administración de evaluaciones de recursos educativos

Caso de uso administración de evaluaciones de recursos educativos	
Actores principales	Usuario con permisos de Administrar evaluaciones de recursos
Permite consultar y dar seguimiento a las Evaluaciones de Recursos educativos mediante la edición del recurso y el cambio de estado de la evaluación.	
Pre-condiciones	Del proceso <ul style="list-style-type: none">▪ El usuario debe contar con su <i>Cuenta de usuario</i> y <i>Contraseña</i>. Del sistema <ul style="list-style-type: none">▪ El <i>usuario</i> debe autenticarse en el sistema.▪ El <i>usuario</i> debe contar con el permiso de administrar evaluaciones de recursos y contar con estatus Activo.▪ El sistema debe tener evaluaciones de recursos registradas.

Flujo Principal	<p>Este Caso de Uso inicia cuando el <i>usuario</i> ingresa al módulo de Evaluación de Recursos.</p> <ol style="list-style-type: none"> 1. El sistema permite al <i>usuario</i> Buscar evaluaciones registradas en el sistema (ver Flujo Alterno A-1 Buscar evaluaciones). 2. El sistema muestra el Listado de evaluaciones (N-1). Los registros aparecen ordenados por Fecha, del más reciente al más antiguo (E-1). Los datos que se muestran en el listado son: <ol style="list-style-type: none"> a) Id. b) Id del recurso. c) Título. Se refiere al título del recurso que a su vez es un enlace al mismo. d) Observaciones. Se muestran las observaciones registradas en el formulario. e) Fecha de evaluación. Fecha en la que se registró la evaluación. f) Usuario que evaluó el recurso. g) Último usuario que dio seguimiento a la evaluación. h) Dictamen. i) Calificación global. j) Estado de seguimiento. <p>El sistema permite ordenar el listado por todos los rubros presentes en el listado a excepción por los campos Observaciones y Plan/Elemento del programa de estudios. Se muestra también el número total de registros que conforman el listado.</p> 3. Por cada registro el sistema permite realizar la siguiente acción: 4. a. Consultar la evaluación del recurso (Ver Flujo Alterno A-2 Consultar evaluación) para ver el detalle de la evaluación y darle seguimiento. <p>Termina el Caso de Uso.</p>
-----------------	---

Flujos alternos

A-1 Consultar Evaluación

1. El usuario selecciona Consultar evaluación del recurso.
2. El sistema muestra la siguiente información:
 - Id del recurso.
 - Título. Es un enlace para ir al recurso si es de tipo URL o para descargar el recurso si es de tipo Archivo. (N-5).
 - Usuario que evaluó el recurso.
 - Dictamen.
 - Fecha de evaluación.
 - Estado de seguimiento.
 - Evaluación del Recurso Educativo Digital RUA. Se muestran todos los datos del formulario conformado por las secciones: Validez, Aspectos técnicos y Dictamen.
 - El sistema permite realizar las siguientes acciones:
 - Editar recurso. El sistema permite modificar la información del recurso educativo.
 - Modificar estado. El sistema permite modificar el estado de seguimiento de una evaluación de “Pendiente” a “Atendido” y viceversa (N-3).
 - Observaciones.
3. El usuario realiza las modificaciones pertinentes y selecciona “Guardar” la información.
4. El sistema valida que la información sea correcta, guarda el seguimiento de la evaluación y envía el siguiente mensaje: “Los cambios se han guardado correctamente” (N-3).
5. El sistema muestra el listado de evaluaciones.

A-2 Buscar evaluaciones

1. El usuario podrá ingresar los siguientes criterios de búsqueda opcionales (N-2):
 - Título del recurso.
 - Id del recurso
 - Usuario que evaluó el recurso.
 - Último usuario que dio seguimiento a la evaluación.
 - Estado de seguimiento.
 - Plan/Elemento del programa de estudios.
 - Calificación global.
 - Dictamen.
 - Fecha de evaluación.
2. El sistema valida la información proporcionada (ver campos en el archivo RUA_T_CamposEvaluacion.xlsx) y en caso de ser válida el sistema muestra los resultados coincidentes en el Listado de evaluaciones (E-2) (N-1), y se presenta también el número total de registros coincidentes.

A-3 Editar recurso

1. El usuario elige la opción Editar recurso.
2. El sistema permite modificar la información del recurso educativo.
3. El usuario puede seleccionar entre las siguientes opciones:

	<ul style="list-style-type: none"> • Regresar a evaluación. Permite regresar a Consultar la Evaluación del recurso educativo (ver flujo alternativo A-2 Consultar evaluación). • Guardar todo. Aplican los mismos términos que en el flujo alternativo A-8 Guardar toda la información del Caso de Uso 10. Administrar Recursos educativos. El sistema guarda la información y muestra nuevamente Consultar la evaluación (ver flujo alternativo A-2 Consultar evaluación) para seguir con el seguimiento de evaluación del recurso educativo. • Publicar. Aplican los mismos términos que en el flujo alternativo A-4 Publicar recurso educativo del Caso de Uso 10. Administrar Recursos educativos. • Quitar publicación. Aplican los mismos términos que en el flujo alternativo Quitar publicación de un recurso educativo.
Flujos de excepción	<p>E-1 Aún no existe información registrada Se muestra el mensaje: “No hay evaluaciones registradas.”.</p> <p>E-2 No existe información coincidente El sistema notifica: “No se encontraron resultados que coincidan con el(los) criterio(s) proporcionado(s).”. El sistema permite modificar el (los) criterio(s).</p> <p>E-3 No se ha realizado alguna modificación, ingreso o eliminación de datos El sistema notifica: “Para guardar la información es necesario que se ingrese o modifique algún dato.”. El sistema permite ingresar y modificar la información.</p>
Post-condiciones	El sistema registra en bitácora la acción, el usuario, la fecha y hora de la operación.

Notas

N-1 Si se modifica el estado de la evaluación a “Atendido” se envía un correo de notificación al usuario que evaluó el recurso. El contenido del correo es el siguiente:

Asunto:

Evaluación de Recursos educativos en la Red Universitaria de Aprendizaje [Nombre de la IES].

Contenido:

Estimado Profesor:

Le notificamos que hemos dado seguimiento a la evaluación que usted realizó al recurso [Nombre del recurso educativo] el día [Fecha en que se realizó la evaluación].

Si se ingresaron observaciones en el Seguimiento de la evaluación, en el correo se agrega lo siguiente:

Con base en su evaluación se realizaron las siguientes observaciones: [Observaciones del “Seguimiento de la evaluación”].

Firma:

Si lo considera pertinente, puede realizar modificaciones a la evaluación accediendo a la opción Modificar evaluación del recurso disponible en la ficha técnica del recurso, o bien, desde la opción Mis evaluaciones disponibles en el menú del portal.

Le agradecemos su colaboración y lo invitamos a seguir participando en el proceso de Evaluación de Recursos.

Atentamente,

El equipo de la RUA [Nombre de la IES].

N-2 El Estado de publicación determina si un recurso se encuentra o no disponible para su consulta a través del portal. Corresponde al dato 10.1 Estado de publicación de la sección Control de la catalogación.

N-3 Si el recurso tiene un solo archivo, el sistema permite la descarga del recurso con su extensión original. Si el recurso tiene más de un archivo, el sistema descarga el recurso como un archivo .zip.

	<p>N-4. El sistema permite exportar a un archivo de Excel el listado de evaluaciones mostradas de dos formas:</p> <ul style="list-style-type: none"> • Listado: se exporta la información mostrada tal cual se presenta en el listado. • Evaluaciones: se exporta la evaluación completa de cada recurso, es decir con la información de cada una de las secciones mostradas en el formulario. <p>N-5. Una vez que se realiza la búsqueda, al mostrar los resultados se ocultan los criterios pero se mantienen seleccionados y/o ingresados.</p>
--	---

A.2. Caso de uso Reporte listado de recursos educativos

Caso de uso listado de recursos educativos	
Actores principales	Usuario con permiso para consultar listado de recursos educativos
Permite consultar un listado con la información de los recursos educativos capturados a partir del ingreso de criterios de búsqueda.	
Pre-condiciones	<p>Del proceso</p> <ul style="list-style-type: none"> ▪ El usuario debe contar con su Cuenta de usuario y Contraseña. <p>Del sistema</p> <ul style="list-style-type: none"> ▪ El usuario debe haberse autenticado previamente con su Cuenta de usuario y Contraseña. ▪ El usuario debe tener permiso para Consultar listado de recursos educativos y tener estatus activo. ▪ Deben existir recursos educativos capturados.

<p>Flujo principal</p>	<p>Este caso de uso se inicia cuando el usuario ingresa al módulo de Reportes.</p> <ol style="list-style-type: none"> 1. El usuario elige Listado de recursos educativos. 2. El sistema permite seleccionar los siguientes criterios para consultar el listado (E-1), salvo en donde se especifique lo contrario todos los criterios son opcionales: <ul style="list-style-type: none"> • Texto. Para este criterio la búsqueda aplica sobre los campos Título y Descripción de recursos. • Valoración. • Autor. • Usuario que creó/importó el recurso. Nombre del usuario que catalogó primero el recurso, o bien importó un recurso RUAMX. • Usuario que realizó la última modificación. Se mostrará la lista de nombres completos de usuarios con perfil de Administrador de recursos educativos. • Plan de estudios. El sistema muestra los planes de estudios que tienen recursos educativos asociados, agrupados por Nivel de estudios (Educación media superior, Técnico superior universitario, Licenciatura, Especialidad, Maestría, Doctorado y Temas de interés). Al seleccionar un plan de estudios, el sistema permite acceder a los elementos del programa de estudios de sus niveles inferiores y seleccionar el nodo de un nivel específico. • Tipo de formato. • Categoría. • Estado del recurso. Dato perteneciente a la sección “2. Ciclo de Vida” de la catalogación del recurso. • Fuente del recurso educativo. • Estado de publicación. • Nivel de acceso.
-------------------------------	--

	<ul style="list-style-type: none">● Fecha de creación. Permite ingresar un periodo de tiempo en el que se capturaron los recursos:<ul style="list-style-type: none">▪ Desde. Fecha de inicio del periodo.▪ Hasta. Fecha final del periodo.● Información a mostrar: El sistema solicita al usuario que se elijan uno o varios datos a mostrar en el listado:<ul style="list-style-type: none">▪ Categoría.▪ Fuente del recurso educativo.▪ Tipo de formato.▪ Detalle de la asociación. Tipo de asociación que tiene el recurso con el plan de estudios.▪ Fecha de creación.▪ Usuario que creó/importó el recurso.▪ Usuario que realizó la última modificación.▪ Estado de publicación. <p>3. Una vez que se han ingresado los criterios y han sido validados, el sistema realiza la consulta y presenta la siguiente información (N-1) (E-2):</p> <ul style="list-style-type: none">▪ Total de registros coincidentes. Número de recursos educativos que coincidan con los criterios ingresados. <p>4. Enseguida se presenta el listado con la siguiente información de acuerdo con los criterios elegidos (N-2):</p> <ul style="list-style-type: none">● Id. Recurso.● Título del recurso.● Categoría.
--	---

	<ul style="list-style-type: none"> ● Fuente del recurso educativo. Si el recurso es URL, se muestra la liga para abrir el vínculo. Si el recurso es Archivo, se muestra la liga "Descargar recurso educativo" para permitir la descarga (N-3). ● Tipo de formato. ● Detalle de la asociación con el plan de estudios. Se presentan agrupadas por: Lo que ya debo saber, Programas de estudio asociados y Para saber más, las rutas que hacen referencia a la asociación del recurso educativo con uno o varios elementos del programa de estudios del plan de estudios especificado. Por ejemplo: "Escuela Nacional Preparatoria / Cuarto año / 1401 Física III". ● Fecha de creación. Se muestra la fecha de registro del recurso educativo dd/mm/aaaa. ● Usuario que creó/importó el recurso. Nombre del usuario que catalogó primero el recurso. ● Usuario que realizó la última modificación. ● Estado de publicación. <p>4. El usuario puede cambiar en cualquier momento los criterios de búsqueda y los datos a mostrar en el reporte.</p> <p>5. El sistema permite ordenar el listado por cada dato mostrado, excepto por Fuente del recurso educativo y Detalle de la asociación.</p> <p>Termina el Caso de Uso.</p>
--	--

<p>Flujos de Excepción</p>	<p>E-1. No se encontraron recursos educativos registrados El sistema notifica: “No hay recursos educativos registrados.”. Termina el caso de uso.</p> <p>E-2. No se encontraron coincidencias El sistema notifica: “No se encontraron resultados que coincidan con el(los) criterio(s) proporcionado(s).”. El sistema permite especificar nuevos criterios de búsqueda.</p>
<p>Notas</p>	<p>N-1. Una vez que se realiza la búsqueda, al mostrar los resultados se conservan los criterios seleccionados y/o ingresados.</p> <p>N-2. El sistema permite exportar a Excel los datos del listado de recursos educativos, con la siguiente consideración: En el caso de que la Fuente del recurso sea archivo, se muestra la leyenda “Archivo”.</p> <p>N-3. En caso de que el recurso contenga más de un archivo, se permite descargar un archivo comprimido .zip que incluye todos los archivos del recurso.</p>

Bibliografía

- [1] Sommerville, I. (2010). *Software Engineering*. (9° ed.). EE. UU.: Addison-Wesley Publishing Company.
- [2] Ibargüengoitia G. G. & Oktaba H. (2010). *Ingeniería de Software Pragmática*. México: Facultad de Ciencias UNAM.
- [3] Equipo de análisis RUA. (2017). *Especificación de requerimientos RUA IES*.
- [4] Equipo de análisis RUA. (2017). *Caso de uso seguimiento de evaluación*. RUAT_25_SeguimientoEvaluación.docx.
- [5] Equipo de análisis RUA. (2017). *Caso de uso listado de recursos educativos*. RUAT_15.ConsultarListadoEvaluacionRecursosEducativos.docx.
- [6] PHP: ¿What is PHP? - Manual. (Septiembre 2018). <http://php.net/manual/en/intro-what-is.php>.
- [7] PHP: History of PHP - Manual. (Septiembre 2018). <http://php.net/manual/en/history.php>.
- [8] CookBook - Welcome 2.X. (Septiembre 2018). <https://book.cakephp.org/2.0/en/getting-started.html>.
- [9] A typical CakePHP Request. (Septiembre 2018). <https://book.cakephp.org/2.0/en/getting-started/a-typical-cakephp-request.html>.
- [10] PostgreSQL: About. (Septiembre 2018). <https://www.postgresql.org/about/>.
- [11] PostgreSQL: Documentation: 9.5: Introduction. (Septiembre 2018). <https://www.postgresql.org/docs/9.5/static/textsearch-intro.html>.
- [12] PostgreSQL: Documentation: 9.5: Text Search Types. (Septiembre 2018). <https://www.postgresql.org/docs/9.5/static/datatype-textsearch.html>.

- [13] PostgreSQL: Documentation: 9.1: WITH Queries (Common Table Expressions). (September 2018). <https://www.postgresql.org/docs/9.1/static/queries-with.html>.
- [14] Historical trends in the usage of server-side programming languages. (October 2018). https://w3techs.com/technologies/history_overview/programming_language.