



TÍTULO PROYECTO

**CATÁLOGO DE SERVICIOS USTA PARA LA PLATAFORMA DE INGLÉS DEL
DEPARTAMENTO DE IDIOMAS DE LA UNIVERSIDAD SANTO TOMÁS SECCIONAL
TUNJA.**

PROPONENTE(S)

Yesica Daniela Vargas Espitia

1049649867

2165503

Andrés Mauricio Pérez Escobar

1049642165

3111411

Andrés Felipe Herrera Rodríguez

1049650247

2168484

DIRECTOR

Edgar Medina Sánchez

Tunja

Fecha de presentación (día, mes, año)

CONTENIDO

1.	FICHA TÉCNICA DEL PROYECTO	7
2.	PLANTEAMIENTO DEL PROBLEMA	8
3.	JUSTIFICACIÓN	10
4.	OBJETIVOS	11
5.	MODELO DE DESARROLLO	12
6.	DESARROLLO DEL PROYECTO	17
6.1.	Fase 1. Análisis de requerimientos	18
6.2.	Fase 2. Introspección de código a nivel de controladores	24
6.3.	Fase 3. Diseño y refinación de requerimientos	32
7.	CONCLUSIONES	72
8.	ANÁLISIS DE RESULTADOS	73
9.	REFERENCIAS	74
10.	ANEXOS	75
	Anexo 1. Entrevista dirigida al encargado de la plataforma	75
	Anexo 2. Plantilla de Requerimientos	77
	Anexo 3. Mapa Conceptual	81
	Anexo 4. Diagrama de Secuencia Cursos Permitidos	82
	Anexo 5. Diagrama de Secuencia Actualizar Curso	83

Lista de Tablas

Tabla 1. Productos de ingeniería y su relación con el ciclo de vida de ingeniería de software	10
Tabla 2. Objetivos específicos	11
Tabla 3. Análisis de Requerimientos.....	18
Tabla 4. Introspección de código a nivel de controladores.....	24
Tabla 5. Fase 3.....	32
Tabla 6. Fase 4.....	51
Tabla 7. Fase 5.....	60
Tabla 8. Caso de uso iniciar sesión	77
Tabla 9. Caso de uso Listar cursos.....	77
Tabla 10. Caso de uso cargar subtemas	78
Tabla 11. Caso de uso Listar actividades	78
Tabla 12. Caso de uso ver contenido temático	79
Tabla 13. Caso de uso ver perfil de usuario.....	79
Tabla 14. Caso de uso ver mis cursos.....	80
Tabla 15. Caso de uso mostrar certificados.....	80

Lista de Figuras

Figura 1. Diagrama de espina de pescado.....	9
Figura 2. Modelo 4+1 vistas.....	14
Figura 3. Pantalla Principal.....	19
Figura 4. Login.....	19
Figura 5. Cursos.....	19
Figura 6. Subtemas.....	20
Figura 7. Actividades.....	21
Figura 8. Contenido.....	21
Figura 9. Mapa conceptual de servicios.....	23
Figura 10. Composición general de la aplicación.....	25
Figura 11. Composición del paquete config.....	25
Figura 12. Composición del paquete modules.....	26
Figura 13. Composición del paquete resources.....	26
Figura 14. Clases paquete cefr.....	29
Figura 15. Clases paquete cefr2.....	29
Figura 16. Clases paquete instituciones.....	30
Figura 17. Clases paquete usermanager.....	31
Figura 18. Diagrama de Clases Autenticar.....	33
Figura 19. Diagrama De Secuencia Autentica.....	34
Figura 20. Diagrama De Actividades Autenticar.....	35
Figura 21. Diagrama Casos de Uso Autenticar.....	35
Figura 22. Diagrama de Clases Cursos Permitidos.....	36
Figura 23. Diagrama De Secuencia Cursos Permitidos.....	37
Figura 24. Diagrama De Actividades Cursos Permitidos.....	37
Figura 25. Diagrama Casos de Uso Cursos Permitidos.....	38
Figura 26. Diagrama de Clases Cursos Permitidos.....	39
Figura 27. Diagrama de Secuencia Cursos Permitidos.....	39
Figura 28. Diagrama De Actividades Cursos Permitidos.....	40
Figura 29. Diagrama Casos de Uso Obtener Certificado.....	40

Figura 30. Diagrama de Clases Actualizar Curso.....	41
Figura 31. Diagrama de Secuencia Actualizar Curso.....	41
Figura 32. Diagrama De Actividades Actualizar Curso.....	42
Figura 33. Diagrama Casos de Uso Actualizar Curso	42
Figura 34. Diagrama de Paquetes	43
Figura 35. Diagrama de Componentes	44
Figura 36. Diagrama de Despliegue	45
Figura 37. Diseño Interfaz Autenticar.....	46
Figura 38. Diseño Interfaz Autorizar.....	46
Figura 39. Diseño Interfaz Listar Cursos.....	47
Figura 40. Diseño Interfaz Subtemas.....	47
Figura 41. Diseño Interfaz Actividades.....	48
Figura 42. Diseño Interfaz Contenido.....	49
Figura 43. Diseño Interfaz Certificado.....	49
Figura 44. Diseño Interfaz Actualizar	50
Figura 45. Diseño Interfaz Perfil.....	50
Figura 46. Modelo Arquitectónico 1	52
Figura 47. Modelo del proyecto.....	53
Figura 48. Servicio JPA	53
Figura 49. JPA persistence.xml	55
Figura 50. JPA pom.xml.....	55
Figura 51. Servicio EJB	56
Figura 52. EJB persistence.xml	57
Figura 53. EJB servicios-orm.xml	57
Figura 54. EJB pom.xml.....	58
Figura 55. Servicios WEB.....	58
Figura 56. WEB pom.xml	59
Figura 57. Servicios EAR.....	59
Figura 58. EAR pom.xml.....	60
Figura 59. ServiciosBean Autenticar	61
Figura 60. ServiciosBean Cascarones	62

Figura 61. ServiciosFachada Cascarones	63
Figura 62. Serviciosmanagedbean Cascarones.....	64
Figura 63. Servicio-web Autenticar	65
Figura 64. Servicio-web Autorizar	66
Figura 65. Servicio-web Obtener certificado	67
Figura 66. Servicios-ear despliegue.....	67
Figura 67. Servicio-web Obtener certificado	68
Figura 68. XML Entrada.....	68
Figura 69. XML Salida	69
Figura 70. Conexión android a servicio Autenticar	70
Figura 71. Conexión Android a servicio Autenticar 2.....	70
Figura 72. Pantalla servicio autenticar entrada	71
Figura 73.Salida del servicio autenticar en consola de Android	71

1. FICHA TÉCNICA DEL PROYECTO

Título	CATÁLOGO DE SERVICIOS USTA PARA LA PLATAFORMA DE INGLÉS DEL DEPARTAMENTO DE IDIOMAS DE LA UNIVERSIDAD SANTO TOMÁS SECCIONAL TUNJA.
Autor (es)	Yesica Daniela Vargas Espitia Andrés Mauricio Pérez Escobar Andrés Felipe Herrera Rodríguez
Director del proyecto	Edgar Medina Sánchez
Palabras claves	Web Services, Diseño, Servicios
Descripción	<p>Este proyecto consiste en realizar un catálogo de servicios para el la plataforma de ingles del departamento de idiomas de la USTA Tunja, en el cual, se definieron los servicios que ofrecerá la plataforma de inglés como apoyo en la formación de los estudiantes que deseen hacer una retroalimentación del nivel A1, para la construcción del catálogo donde se utilizó el modelo 4 + 1 vistas, el lenguaje JAVA y XML para la presentación de las API's.</p>

2. PLANTEAMIENTO DEL PROBLEMA

En el departamento de Idiomas de la Universidad Santo Tomás Seccional Tunja, ofrece el curso de inglés con el fin satisfacer las dificultades de la comunidad tomasina que decida tomar este curso, para fortalecer su conocimiento por medio del manejo de tecnologías y software; con la construcción de la plataforma Web, se quiere hacer un seguimiento del avance de estudiantes del nivel A1, sin embargo, la necesidad del departamento de Idiomas en conjunto con la facultad de Ingeniería de Sistemas es ampliar la plataforma para que no sea únicamente web sino que también sea móvil, y así fortalecer la interacción usuario versus curso, por medio del tiempo dedicado a su estudio de cada módulo, realizando consultas permanentes y dando la posibilidad de practicar en cada momento que el usuario desee.

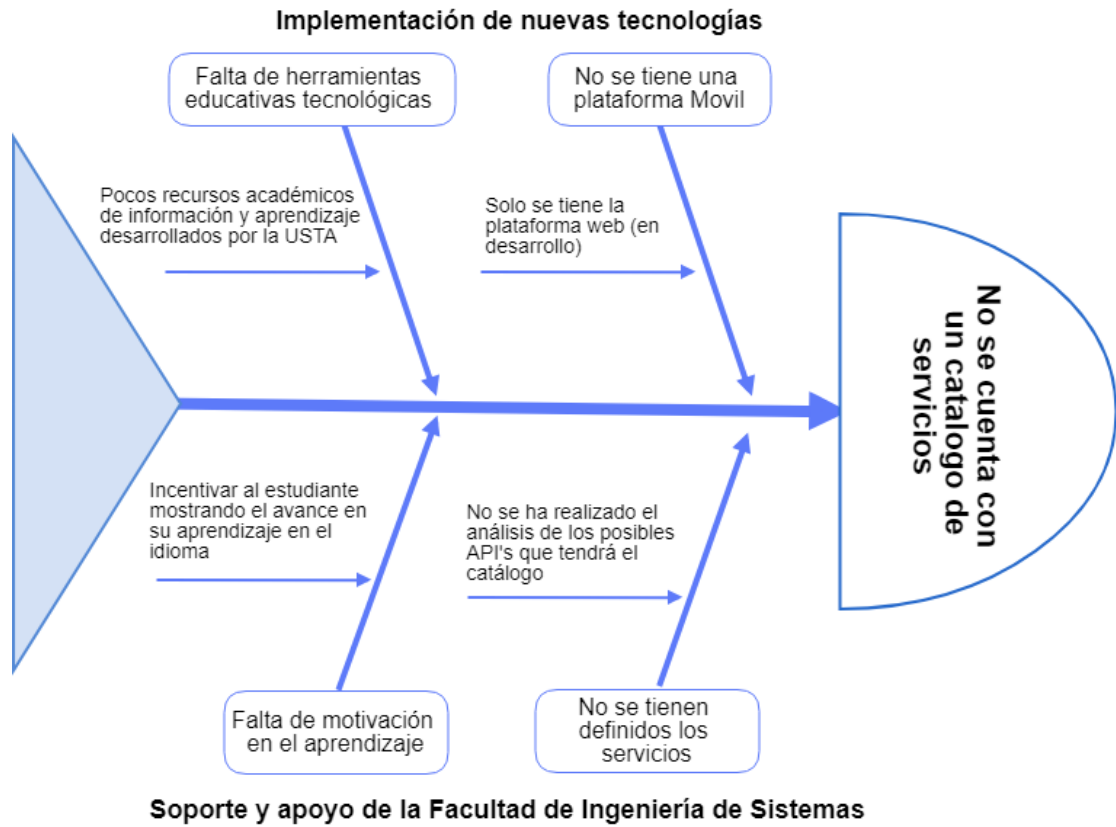
Para construir la aplicación Móvil es necesario diseñar y especificar los servicios web que se requieren de la plataforma Web, los cuales serán desarrollados e implementados a partir de este trabajo. Por lo tanto, se requiere la construcción de un catálogo, que permita presentar los servicios que la plataforma web ofrecerá a los desarrolladores de las aplicaciones móviles, éstas conformarán un sistema el cual facilitará a los docentes, administrativos y estudiantes llevar un control de su progreso.

¿Por qué y para qué un catálogo? Porque es importante definir cuáles serían la API's que serían requeridas por el desarrollador móvil, y con este insumo poder desarrollar las aplicaciones.

En este punto, es importante resaltar la construcción del diseño de cada uno de los servicios ofrecidos en el catálogo, con la definición del servicio y el diseño de este, un desarrollador podrá construir las aplicaciones móviles que se requieren en la plataforma de inglés.

A continuación, se presenta en la figura 1, el diagrama de espina de pescado con la representación de la problemática presentada en el proyecto:

Figura 1. Diagrama de espina de pescado.



Fuente: Autores

3. JUSTIFICACIÓN

El diseño y arquitectura de los servicios nos sirve para mostrar los conceptos aprendidos en la carrera, generar un estándar de programación, implementar y utilizar nuevas tecnologías de desarrollo, facilitar el ciclo de vida del software y la integración entre diferentes lenguajes.

Con la implementación de esta arquitectura se beneficia a la facultad, ya que se deja un estándar previo hacia los desarrolladores, así facilitarles la comprensión del mismo y del funcionamiento interno de los servicios expuestos, con el fin de disminuir los tiempos del desarrollo.

Este modelo arquitectónico propuesto, es exclusivo para creación de servicios web, se utilizó java, para implementar y fortalecer los conocimientos adquiridos en las materias de arquitectura empresarial y Web Services, además es más fácil la integración entre el móvil y Java, ya que Android soporta el lenguaje Java. Finalmente se usó el protocolo SOAP ya que nos brinda más seguridad por medio de los WSDL y el transporte de datos que van a estar encriptados por los XML anteriormente generados, con el fin de que la arquitectura de software integre la aplicación mediante los servicios.

Tabla 1. Productos de ingeniería y su relación con el ciclo de vida de ingeniería de software

Estudiante	Producto por entregar	Ciclo de vida ingeniería de software
Andrés Pérez	Catálogo de servicios con las API's especificadas	Análisis de requerimientos Introspección de código a nivel de controladores Despliegue del catálogo
Andrés Herrera	El diseño de los servicios soportado en el modelo de vistas de arquitectura 4 + 1 propuesto por Philippe Kruchten	Diseño y refinación de requerimientos Pruebas de diseño
Yesica Vargas	Maquetado y cascarón de los servicios en Java Enterprise Edition	Integración del diseño arquitectónico y catálogo de servicios.

Fuente. Autores.

4. OBJETIVOS

4.1. Objetivo General

Construir el catálogo de servicios que defina las API's que ofrecerá la plataforma de inglés para el apoyo en la formación del nivel A1, utilizando Java como lenguaje para la construcción del catálogo y XML para la presentación de las API's, tomando como referencia el modelo propuesto por Philippe Kruchten para la construcción de los diseños de los servicios.

4.2. Objetivos específicos

La descripción de los objetivos específicos se puede apreciar en la Tabla 2

Tabla 2. Objetivos específicos

Nro.	Objetivo específico
1	Analizar la plataforma de inglés interactuando con los servicios web ofrecidos por la plataforma, para construir un mapa conceptual que sirva de base en la construcción del catálogo.
2	Realizar la introspección del código fuente de la plataforma de inglés, con el fin de definir los servicios que harán parte del catálogo, garantizando la reutilización del código realizado en la plataforma Web.
3	Diseñar los servicios definidos en el catálogo utilizando el modelo de arquitectura 4 + 1 vistas, para que los desarrolladores de las aplicaciones móviles tengan un insumo que les permita comprender el funcionamiento internos del servicio expuesto.
4	Construir el cascarón de los servicios a exponer, de tal manera que los desarrolladores de la plataforma Web puedan tener una línea de base para la construcción de los servicios.
5	Realizar el despliegue del catálogo dentro del componente administrador de la plataforma de inglés, para que los desarrolladores puedan tener acceso a todo el catálogo.

Fuente: Autor

5. MODELO DE DESARROLLO

Este modelo de desarrollo se elaboró con el fin de generar los diseños de los servicios que se identificarán en el desarrollo de este trabajo, tomando como base el modelo que se describirá continuación:

- Se elaboró la vista de escenarios donde se realizaron los diagramas de casos de uso, resumiendo quién podrá usar la plataforma y qué acciones pueden hacer.
- Con la vista lógica, se toma como en el desarrollo del modelo, donde se identifican las clases que se usarían en cada servicio y así poder representar la estructura de la plataforma; luego se elaborará el diagrama de secuencias donde se evidencia los procesos y sus objetos que hacen parte del flujo de los servicios.
- Posteriormente se realizó la vista de despliegue, el cual está conformado por el diagrama de componentes, que permite visualizar la estructura de alto nivel del sistema y el comportamiento de los servicios de la plataforma; también hace parte el diagrama de paquetes, que nos ayuda a tener una visión más clara de los objetos, agrupados en subsistemas y detallando las relaciones de dependencia entre ellos.
- Siguiendo el siguiente paso se elaboró la vista de procesos, diseñando el diagrama de actividades, en el cual se observa un flujo de procesos que se ejecutarán en los servicios de la plataforma.
- Por último, se desarrolló la vista física, que tiene el diagrama de despliegue, que permite modelar la arquitectura en tiempo de ejecución y muestra la configuración de los elementos de hardware de la plataforma.

Para el desarrollo de este proyecto, se contemplan cinco fases distribuidas de la siguiente manera:

Fase 1-Levantamiento y Análisis de requerimientos: En esta fase se trabajó el modelo de análisis y levantamiento de requisitos, teniendo en cuenta las buenas prácticas para el desarrollo de especificación de requerimientos de software (LOPEZ, PEDRO, 2015).

Se desarrollaron las siguientes actividades según las buenas prácticas encontradas y clasificadas según las necesidades encontradas.

Actividades:

- Entrevista con encargados del software en la actualidad.
- Análisis de la plataforma
- Mapa conceptual de la plataforma.

Fase 2-Introspección de código a nivel de controladores: En esta fase se realizó la introspección del código facilitado por el encargado de la plataforma, teniendo en cuenta las sugerencias y las entrevistas realizadas con él, además de esto para la obtención y estudio del código fuente se realizaron las siguientes 2 actividades las cuales se describen a continuación.

Actividades:

- Analizar el código fuente: identificación y definición de la arquitectura, paquetes, clases, archivos de propiedades, lenguaje de programación en el cual se está desarrollando la aplicación, y el servidor se está desplegando para conocer restricciones a nivel de código fuente, captura de errores, de esta manera poder hacer una primera clasificación de los servicios que se ofrecerán para evidenciarlos en el catálogo. Hacer uso de las clases modelo de la aplicación para la construcción del catálogo.

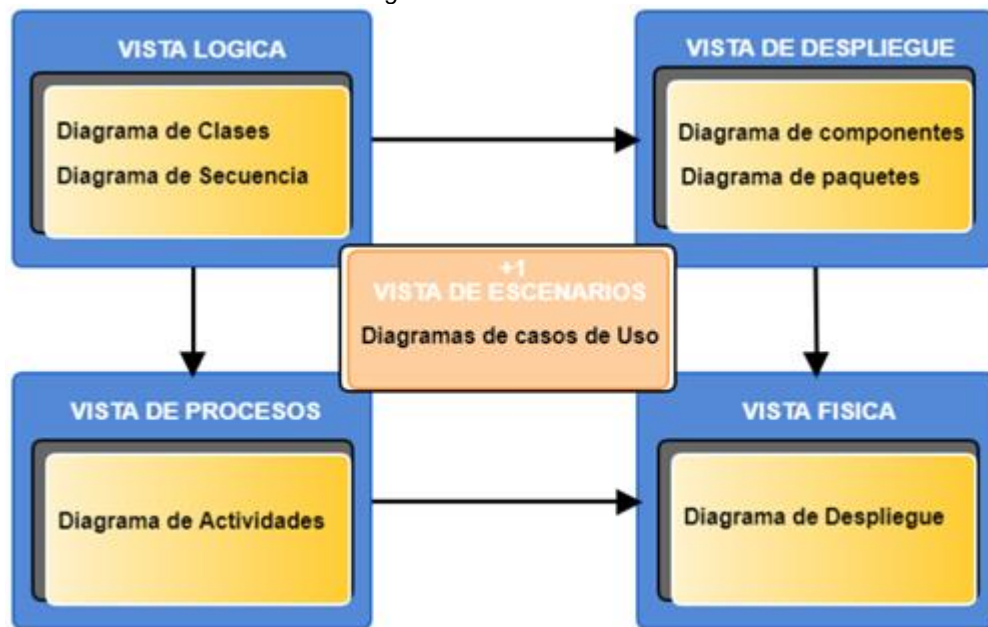
Fase 3-Diseño y refinación de requerimientos: Se realizó toma como referencia el modelo de 4+1 vistas propuesto por Philippe Kruchten [1], con el cual se puede evidenciar la estructura del catálogo de servicios, definiendo así que arquitectura será la más

apropiada de utilizar para el proceso de desarrollo, determinando de esta manera los componentes que fueron involucrados.

Actividades:

- Descripción de modelo 4+1 Vista: Según Kruchten, se aconseja usar un modelo de 4+1 vistas, para el desarrollo de aplicaciones móviles y web, porque este modelo puede responder a las distintas partes interesadas en este proyecto, programadores, clientes, administradores entre otros, se presenta una visión resumida del sistema con la información que requieren para satisfacer sus necesidades.

Figura 2. Modelo 4+1 vistas



Fuente: Autores

Capa lógica: En la capa lógica se verá representada la funcionalidad que el sistema le brindará al usuario, en esta se representará la interacción de los objetos encontrados, para que la documentación esté completa se incluyen los diagramas de secuencia y de clases de UML.

Vista de Despliegue: Aquí se mostrará el sistema desde la vista del programador y se encargará de la gestión de software, es decir, se va a mostrar cómo está dividido el sistema, observando los componentes y las dependencias que hay en esos componentes, para que la documentación esté completa se incluyen los diagramas de componentes y de paquetes de UML.

Vista de Procesos: Se mostrarán los procesos que están involucrados en el sistema, se representará la comunicación desde el punto de vista de un integrador de sistemas, para la complementación de esta vista se incluirá el diagrama de actividades en UML.

Vista Física: Se mostrará desde el punto perspectivo del ingeniero de sistemas los componentes físicos que tiene el sistema, las conexiones físicas que existen entre los componentes que están conformando este sistema también se incluyen los servicios, para que esta documentación quede completa se incluirá el diagrama de despliegue en UML.

“+1” Vista de Escenarios: En Esta vista se unirán las otras cuatro vistas ya mencionadas y se representará por casos de uso y así ver cómo interactúan los diagramas, con lo que tendremos una trazabilidad de componentes, clases, equipos, paquetes, etc., para realizar cada caso de uso. Para la documentación se debe incluir los casos de uso de UML.

Fase 4 – Desarrollo de proyecto: En esta fase se evidencia la puesta en práctica de cada una de la capas de la fase anterior, conformado por las actividades propuestas por el modelo 4+1 vistas en cuanto la creación de modelos UML (Diagramas de casos de Uso, Diagrama de Clases, Diagrama de Secuencia, Diagrama de Paquetes, Diagrama de Componentes, Diagrama de Actividades, Diagrama de Despliegue), modelo arquitectónico propuesto (Eclipse, maven, netbeans, jboss, mysql, cacao, ejb, ear, jpa, soapUi).

Arquitectura del Servicio Web

Un web service es un conjunto de protocolos el cual nos ayuda a comunicar datos o información, entre varias aplicaciones, ya sean web y móvil, por medio de la creación de los servicios necesarios, con sus respectivos estándares.

En este caso se usó el protocolo SOAP (Simple Object Access Protocol), este es la base de los Web Services y nos permite tener comunicación entre las aplicaciones por medio de Internet o localmente.

También fue necesario la utilización del protocolo WSDL (Web Services Description Language), el cual está basado en XML y es el que nos otorga el acceso al Web Service. Es muy importante ya que es el que nos dice qué tipos de datos se están utilizando en el servicio.

Herramientas que se utilizaron en el desarrollo del proyecto:

Eclipse: Es un software de código abierto, que nos permite desarrollar en entornos de desarrollo integrado (Integrated development environment), el cual contiene numerosos componentes de herramientas, para la programación de los proyectos a realizar. [1]

MAVEN: Es un componente libre que viene integrado en Eclipse, el cual nos ayuda a simplificar los procesos de construcción, como lo puede ser realizar una serie de tareas establecidas, compilación de código y empaquetamiento. [2]

NETBEANS: Es un entorno de desarrollo integrado, el cual nos permite programar en varios lenguajes de programación y nos facilita el uso de varias tecnologías para el desarrollo de aplicaciones, en este caso se utilizó para realizar la introspección de código del proyecto web. [3]

JBoss: Es un servidor de aplicaciones implementado en el lenguaje de programación Java EE con código abierto integrado en Eclipse, este servidor se utilizó para subir y exponer los servicios a utilizar. [4]

MySQL: Es un motor de base de datos relacional, el cual maneja un lenguaje de consulta estructural y es de uso libre o comercial. [5]

Cacoo: Es una herramienta online el cual nos sirve de base para realizar varios tipos de diagramas, por medio de plantillas ya establecidas, en este caso se utilizó para elaborar los diagramas UML del modelo 4+1 vistas. [6]

EJB: Es una clase que tiene la anotación @EJB, la cual se encarga de realizar los diferentes métodos programados en la base de datos, dependiendo del JPA. [7]

EAR: Es un archivo que empaqueta los jpa, ejb y servicios web de un proyecto, este archivo es el que se sube al servidor de aplicaciones cada vez que se hace un despliegue. [7]

JPA: Es una API de persistencia del lenguaje de programación Java, la cual manipula datos relacionales en las aplicaciones. En este caso se utilizó para hacer el mapeo de las entidades que se tenían en la base de datos. [7]

SoapUi: Es un software basado en java, que sirve para la ejecución de pruebas de los servicios creados, en este caso se encarga de comprobar el funcionamiento de la aplicación siempre y cuando tenga una arquitectura orientada a servicio (SOA). [8]

Actividades: Según el modelo 4+1 vistas [1] se plantearon los diseños de los siguientes modelos los cuales facilitarán la documentación para los desarrolladores de los servicios.

Realizar la construcción del maquetado de los servicios de exposición y consumo en aplicación Android que llevará la plataforma, mediante pruebas de la implementación.

Fase 5-Despliegue del catálogo: para esta fase se proponen realizar dos actividades las cuales son: ensamblar el catálogo con su maquetado y cascarones de los servicios a ofrecer y desplegar el catálogo dentro del componente administrador de la plataforma de inglés, las cuales serán la columna principal de la entrega del catálogo.

6. DESARROLLO DEL PROYECTO

En esta sección, se detalla cada una de las etapas realizadas para el desarrollo del proyecto, teniendo en cuenta las siguientes fases:

6.1. Fase 1. Análisis de requerimientos

En esta fase, se realizó la entrevista con el ingeniero responsable de la plataforma web, con la finalidad de comprender aspectos del sistema relacionados con la confidencialidad, rendimiento, soporte de la aplicación, entre otros. Adicionalmente se recolectaron productos como el código fuente, documento de requisitos, diseño web y demás información para poder conocerla, en la Tabla 3 se observa los resultados obtenidos en el desarrollo de la entrevista con el administrador.

Tabla 3. Análisis de Requerimientos

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> ● Entrevista realizada
Instrumentos y/o herramientas	<ul style="list-style-type: none"> ● Sitio de trabajo ● Conexión a internet ● Computador ● Programas del computador <ul style="list-style-type: none"> ✓ Navegador Chrome ✓ Word ✓ Cacao
Salidas	<ul style="list-style-type: none"> ● Código Interfaces ● Fuente ● Requisitos Funcionales ● Mapa Conceptual
Tiempo	Cuatro semanas

Fuente: Autor

- Solicitud de la plataforma de inglés

Entrevista: Para conocer el estado actual y la debida información de la plataforma, se realizó una entrevista como una buena práctica del levantamiento de requisitos, con el fin de tener más conocimiento sobre la misma. Remítase al [Anexo 1](#). Entrevista dirigida al encargado de la plataforma.

Diagnóstico: Según la recopilación de la encuesta más la entrevista se evidencia que la plataforma está en construcción, sin embargo, se entregaron las imágenes de las interfaces que tiene la plataforma y el código fuente que han realizado, donde finalmente se obtendrán los flujos de los procesos de la aplicación que se está desarrollando.

Para esta primera fase se entrevistó al encargado de la plataforma de inglés, donde se solicitó respetuosamente una copia del producto existente, para poder interactuar y obtener más información sobre la plataforma. El funcionario amablemente aclaró las inquietudes con la entrevista realizada, nos envió la aplicación vía correo electrónico y nos explicó que la plataforma web es una herramienta educativa para los estudiantes de esta institución, la cual ofrece una serie de cursos, subtemas, lista de actividades y contenidos del nivel A1, como base para el aprendizaje del idioma inglés.

A continuación, se podrá evidenciar algunas de las imágenes de la aplicación.

Figura 3. Pantalla Principal.



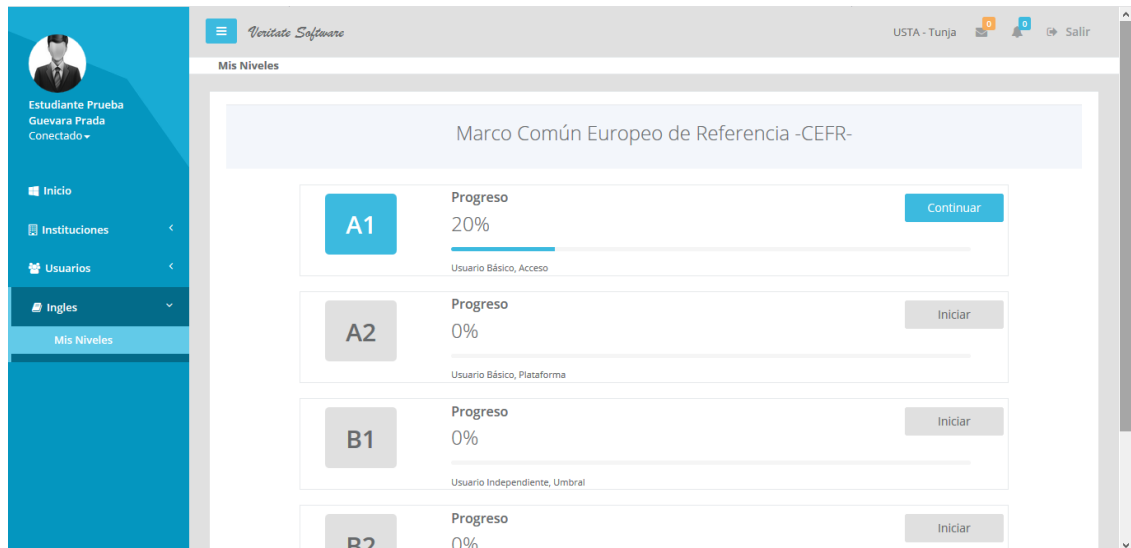
Fuente. Ing. Carlos Andrés Guerrero.

Figura 4. Login.



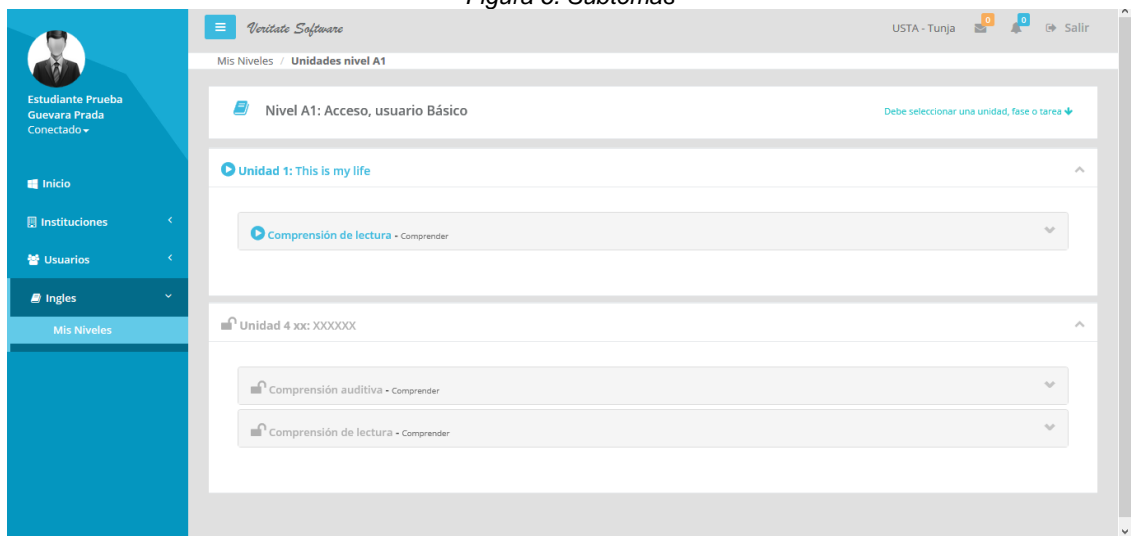
Fuente. Ing. Carlos Andrés Guerrero.

Figura 5. Cursos



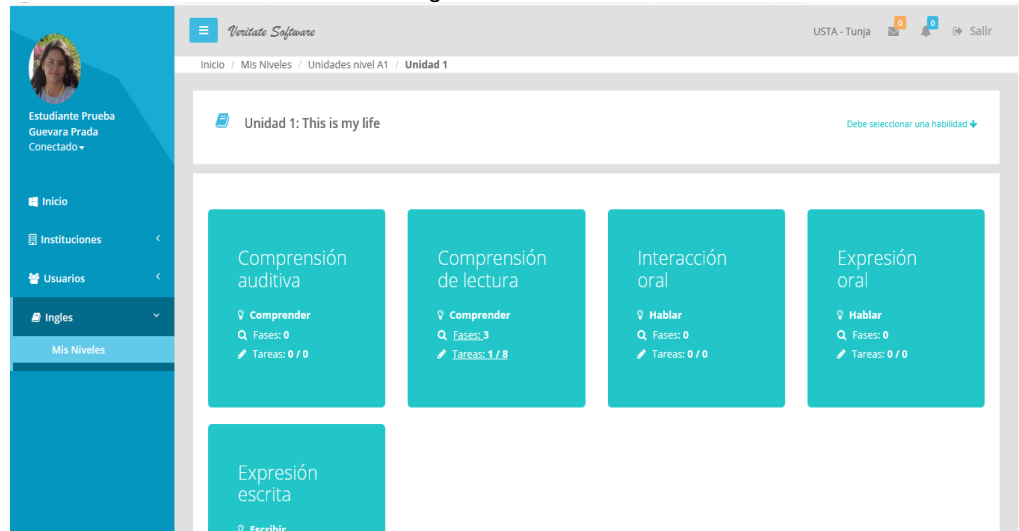
Fuente. Ing. Carlos Andrés Guerrero.

Figura 6. Subtemas



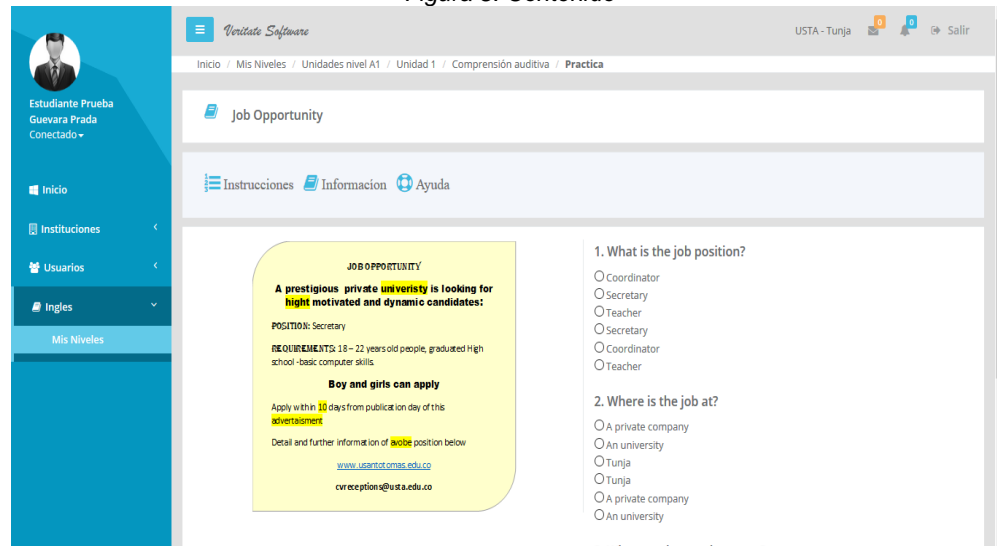
Fuente. Ing. Carlos Andrés Guerrero.

Figura 7. Actividades



Fuente. Ing. Carlos Andrés Guerrero.

Figura 8. Contenido



Fuente. Ing. Carlos Andrés Guerrero.

- Interacción con la plataforma

Una vez el correo fue recibido, se descargó el archivo para poder hacer uso de la plataforma, observar su estructura e identificar los servicios que ofrece.

A continuación, se podrá observar los servicios brindados por la plataforma.

- a. **Inicio Sesión:** en este servicio, el usuario podrá ingresar a la plataforma con su respectivo usuario y contraseña (correo electrónico inscrito). Ver anexo 2 Plantilla de Requerimientos CU001 (Iniciar Sesión).
- b. **Listar Cursos:** una vez el usuario accede a la plataforma, podrá observar la lista de los cursos del idioma inglés, según el nivel avanzado. Ver anexo 2 Plantilla de Requerimientos CU002.1 (Listar Cursos).
- c. **Cargar Subtemas:** cuando el usuario ingrese a la plataforma y seleccione el curso, observará la lista de subtemas, ejemplo: THIS IS MY LIFE, THIS IS MY FAMILY & LISTEN. Ver anexo 2 Plantilla de Requerimientos CU002.1.1 (Cargar Subtemas).
- d. **Listar Actividades:** después de que el usuario elige el subtema, se carga la lista de actividades a desarrollar, como, por ejemplo: PHASES OR ACTIVITIES. Ver anexo 2 Plantilla de Requerimientos CU002.1.1.1 (Listar Actividades).
- e. **Ver Contenido Temático:** cuando el usuario elige la actividad, podrá visualizar el contenido, las tareas y las prácticas a desarrollar. Ver anexo 2 Plantilla de Requerimientos CU002.1.1.1.1 (Ver Contenido Temático).
- f. **Ver Perfil de usuario:** cuando el usuario esté registrado y logueado, podrá acceder a la información de su perfil, para poder actualizar o editar sus datos personales. Ver anexo 2 Plantilla de Requerimientos CU003.1 (Ver Perfil de usuario).
- g. **Ver Mis Cursos:** para este caso de uso, el usuario podrá observar qué cursos ha finalizado y cuales están en proceso. Ver anexo 2 Plantilla de Requerimientos CU004.1 (Ver Mis Cursos).

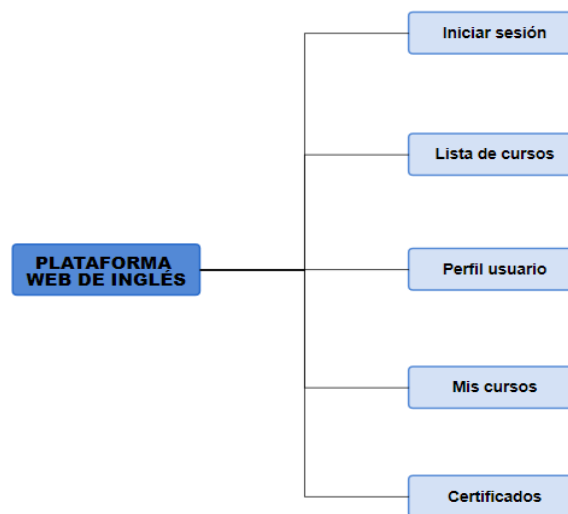
- h. Mostrar certificados:** el usuario podrá observar qué certificados ha obtenido según la terminación y aprobación de los cursos realizados. Ver anexo 2 Plantilla de Requerimientos CU005.1 (Mostrar certificados).

Mapa Conceptual

Este mapa conceptual visualiza los servicios identificados en la plataforma, y según el flujo que tiene, además se explicará cada uno de estos.

Cada una de las fases descritas en este mapa conceptual permitirá detectar el flujo que se debe seguir para la interacción con la plataforma y los resultados que se van obteniendo al paso de cada actividad desarrollada.

Figura 9. Mapa conceptual de servicios



Fuente. Autores

Para mayor detalle dirigirse al Anexo 3 Mapa conceptual, en el cual se encontrará mejor especificación del cuadro.

- **Descripción de servicios según el mapa conceptual**

En este apartado encontramos la composición de los servicios:

- **Iniciar sesión**

- La aplicación contará con un servicio de login, para que el usuario que se autenticó pueda ingresar a la plataforma.
- **Lista de cursos**
 - La aplicación contará con un servicio de listar cursos en el que contienen los subtemas de cada uno, actividades y todo el contenido temático.
 - Para este listado de cursos se cargará la unidad, en la cual se ofrecen los servicios como por ejemplo (Reading, Writing, Speaking) y en cada uno se visualiza sus respectivas actividades.
- **Perfil de Usuario**
 - Se contará con un servicio que presentará la información y tendrá la opción de actualizarla.
- **Mis Cursos**
 - Este servicio mostrará el progreso de los cursos, guardando la sesión para cuando el usuario vuelva a ingresar.
- **Certificados**
 - El servicio otorgará al usuario un certificado que demuestre haber cursado las unidades.

6.2. Fase 2. Introspección de código a nivel de controladores

Para el desarrollo de esta fase, se analizó el código fuente suministrado por el encargado, identificando y definiendo los servicios que se expondrán en el catálogo.

Tabla 4. Introspección de código a nivel de controladores

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> ● Código fuente ● Interfaces
Instrumentos y/o herramientas	<ul style="list-style-type: none"> ● Sitio de trabajo ● Conexión a internet ● Computador ● Programas del computador <ul style="list-style-type: none"> ✓ Navegador Web ✓ Word ✓ NetBeans 8.2
Salidas	Identificación y definición de los servicios que se presentará en el catálogo, reutilización de fragmentos de código (DTO'S) para su desarrollo.
Tiempo	Seis Semanas.

Fuente: Autor

- **Arquitectura de la plataforma web**

En este apartado se encuentra la distribución arquitectónica de cada uno de los componentes que conforman la plataforma.

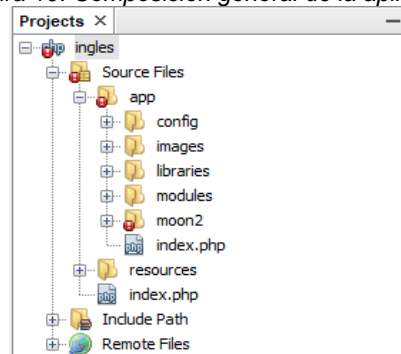
Composición General:

Esta plataforma web se está desarrollando con el lenguaje de programación PHP y está conformado por los siguientes paquetes:

1. Configuración
2. Imágenes
3. Librerías
4. Módulos
5. Recursos adicionales.

En la figura 10 se observa la composición descrita anteriormente.

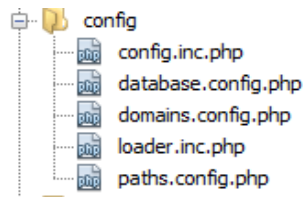
Figura 10. Composición general de la aplicación



Fuente. Ing. Carlos Andrés Guerrero.

El paquete config está compuesto por clases, las cuales permiten la configuración y la parametrización de la aplicación como pueden ser, la conexión con la base de datos, servidor, rutas de acceso al directorio local, gestor documental y demás permisos requeridos por esta.

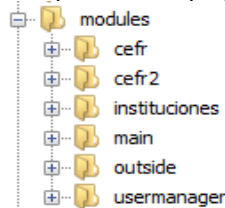
Figura 11. Composición del paquete config



Fuente. Ing. Carlos Andrés Guerrero.

En la carpeta modules, está conformada por paquetes que contienen las clases lógicas con sus respectivos atributos y métodos que describen los objetos de los que está conformada la plataforma.

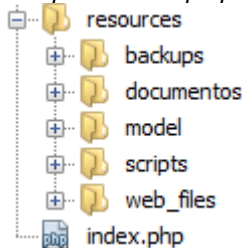
Figura 12. Composición del paquete modules



Fuente. Ing. Carlos Andrés Guerrero.

En resources, se encuentra la documentación adicional y los scripts que se usan para la aplicación web.

Figura 13. Composición del paquete resources



Fuente. Ing. Carlos Andrés Guerrero.

Como resultado de la introspección del código fuente se procede a identificar los objetos que formarán parte de los servicios que se nombrarán en el catálogo.

- **Identificación y Definición de los servicios que ofrecerá el catálogo**

Después de haber realizado la introspección del código, un análisis de cada uno de sus paquetes, clases y archivos de propiedades se identificó los siguientes servicios que harán parte del catálogo:

1. Autenticar

- Descripción del servicio:
 - Este servicio tiene como parámetros de entrada dos cadenas de texto, el nombre de usuario y una contraseña que se cifrará por medio del algoritmo hash (SHA-512), después se hará una comparación de los datos ingresados, con los que están almacenados en la base de datos, donde finalmente retornará un identificador el cual permitirá el acceso a la plataforma.

2. Autorizar

- Descripción del servicio:
 - Verificará el identificador obtenido cuando el usuario ingrese los datos a la plataforma, permitiendo el acceso al módulo de la plataforma con sus respectivos permisos, obteniendo de esta manera la lista de las temáticas.

3. Obtener cursos

- Descripción del servicio:
 - El servicio cargará la lista de los cursos disponibles al usuario por medio del identificador obtenido en del servicio de logeo y que fueron definidos por el administrador.

4. Obtener los subtemas de la temática

- Descripción del servicio:
 - El servicio listará los módulos de los subtemas del curso elegido por el usuario (cliente).

5. Obtener lista de actividades del subtema elegido

- Descripción del servicio:
 - El servicio proveerá los contenidos de las actividades del subtema seleccionado del servicio 4.

6. Obtener el Contenido de aprendizaje

- Descripción del servicio:
 - Este servicio mostrará el contenido de cada actividad perteneciente al subtema seleccionado, estará conformado por la parte gramatical de las materias que pondrá a prueba los conocimientos adquiridos por el usuario (cliente).

7. Obtener Certificado de aprendizaje

- Descripción del servicio
 - Este servicio recibirá una cadena de texto, conformada por código de inscripción del estudiante, código de la persona, nivel cursado, primer nombre y el primer apellido (“12-2168454-1-jose-cuervo”) la cual confirmará que ha terminado satisfactoriamente las actividades por cada temática y retorna un objeto en formato pdf.

8. Actualizar resultado

- Descripción del servicio:
 - Este servicio enviará el objeto Curso donde guardará el avance realizado y retorna el objeto modificado.

9. Actualizar Perfil

- Descripción del servicio:
 - Este servicio mostrará toda la información del usuario registrado, este podrá acceder a los datos contenidos en su perfil, para poder actualizar o editarlos.

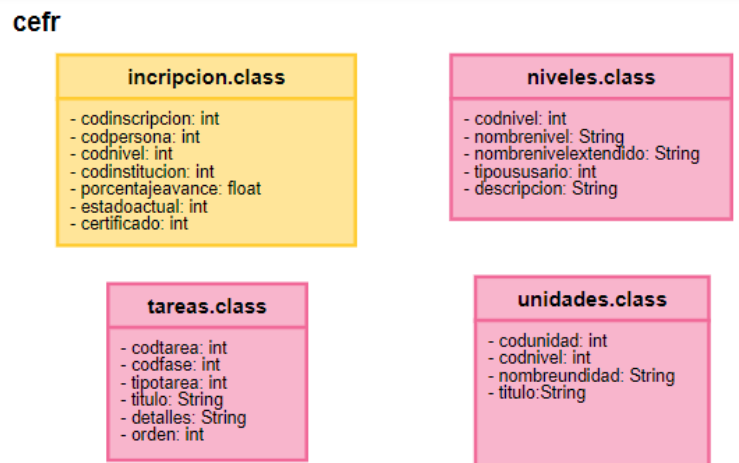
- Código fuente para la construcción del catálogo.

Para la introspección del código fuente lo primero que realizamos fue identificar los módulos, las clases, y los atributos de cada uno de estos, con el fin de agrupar las clases que se tienen en común y así poder crear DTO's (Data Transfer Object) de acuerdo a su sección. En estas figuras se puede observar su agrupación por colores donde se

establecieron tres DTO necesarios para el la construcción del catálogo de servicios, a continuación, observaremos su agrupación por paquetes.

A continuación, se presentan en color rosado las clases que formarán parte del DTO “*CursosVerDTO*”, las clases de color amarillo representaran el DTO “*CursosPermisosDTO*” y los de color azul son “*UsuariosDTO*”, esto con el fin de crear objetos para facilitar la transferencia de los datos de la plataforma.

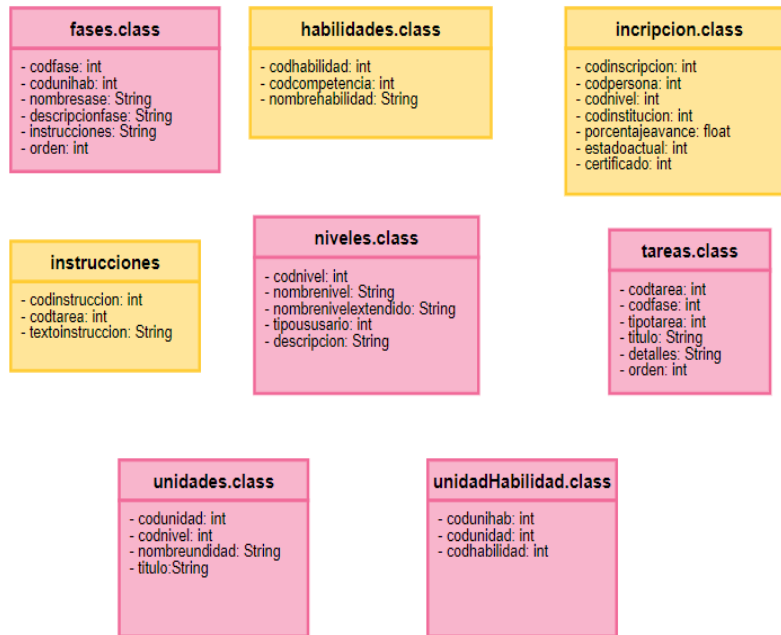
Figura 14. Clases paquete cefr



Fuente. Autores

Figura 15. Clases paquete cefr2

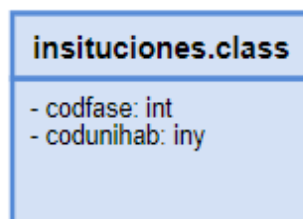
cefr2



Fuente. Autores

Figura 16. Clases paquete instituciones

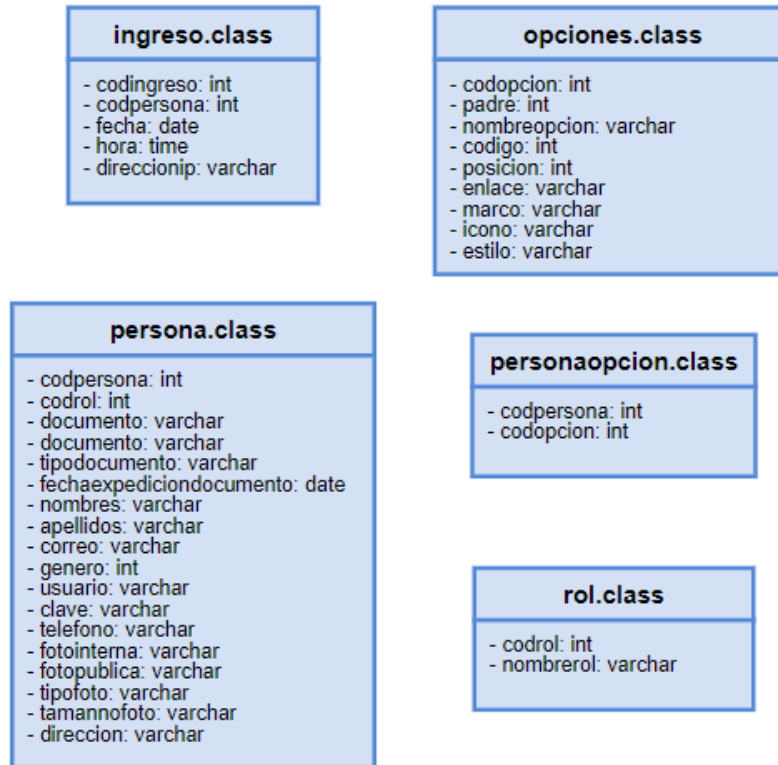
Instituciones



Fuente. Autores

Figura 17. Clases paquete usermanager

usermanager



Fuente. Autores

Los DTO que se establecieron de acuerdo a la información obtenida fueron los siguientes:

1. CursosPermisosDTO

- Inscripciones
- Instrucciones
- Habilidades

2. CursosVerDTO

- Niveles
- Unidades
- UnidadHabilidad
- Tareas
- Fases

3. UsuariosDTO

- Persona
- Ingreso
- Opciones
- Rol
- PersonaOpcion
- Instituciones

6.3. Fase 3. Diseño y refinación de requerimientos

En esta fase se realizó el diseño del modelo 4+1 de los servicios anteriormente propuestos, el cual fue necesario elaborar los diagramas que comprenden este modelo, con el fin de realizar pruebas de las mismas y así dar soporte del desarrollo de la aplicación.

Tabla 5. Fase 3

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> ● Identificación de Requisitos ● Definición de los servicios según requisitos ● Creación código (DTO'S).
Instrumentos y/o herramientas	<ul style="list-style-type: none"> ● Sitio de trabajo ● Conexión a internet ● Computador ● Programas del computador <li style="padding-left: 20px;">✓ Navegador Web <li style="padding-left: 20px;">✓ Word <li style="padding-left: 20px;">✓ Cacao <li style="padding-left: 20px;">✓ lucidchart
Salidas	Diseño del modelo 4 + 1 vistas de los servicios.
Tiempo	Cinco Semanas

Fuente: Autor

Luego de indagar e identificar los diagramas que se utilizaron del modelo 4+1 vistas propuesto por Philippe Kruchten [9], se describió la arquitectura de la plataforma, así realizando una retroalimentación del mismo en la sección 5. *MODELO DE DESARROLLO*, con el fin de la elaboración de los diseños de los servicios anteriormente identificados.

Para el desarrollo de esta fase, la primera vista que se realizó fue la vista lógica, donde se identificaron las clases que se usarían en cada servicio y así poder representar la estructura

de la plataforma; luego se elaboró el diagrama de secuencias donde se evidencian los procesos y sus objetos que hacen parte del flujo de los servicios.

Posteriormente se realizó la vista de despliegue, el cual está conformado por el diagrama de componentes, que permite visualizar la estructura de alto nivel del sistema y el comportamiento de los servicios de la plataforma; también hace parte el diagrama de paquetes, que nos ayuda a tener una visión más clara de los objetos, agrupándolos en subsistemas y detallando las relaciones de dependencia entre ellos.

Después se elaboró la vista de procesos, donde se realizó el diagrama de actividades, en el cual se observa un flujo de procesos que se ejecutarán en los servicios de la plataforma.

Luego, se desarrolló la vista física, que tiene el diagrama de despliegue, que permite modelar la arquitectura en tiempo de ejecución y muestra la configuración de los elementos de hardware de la plataforma.

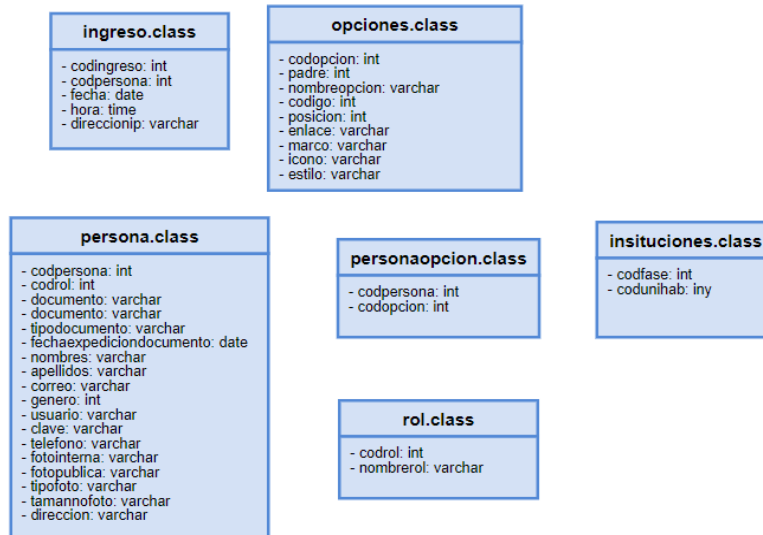
Por último, se elaboró la vista de escenarios donde se realizaron los diagramas de casos de uso, donde se resume las acciones, actores que se ven involucrados y permisos.

Diseño servicios del catálogo.

1. Autenticar

1.1 Diagrama de Clases: En este servicio se usaron cinco clases, las cuales fueron tomadas de la plataforma web que está en construcción, para su debido funcionamiento. A continuación, se puede observar en la siguiente figura:

Figura 18. Diagrama de Clases Autenticar

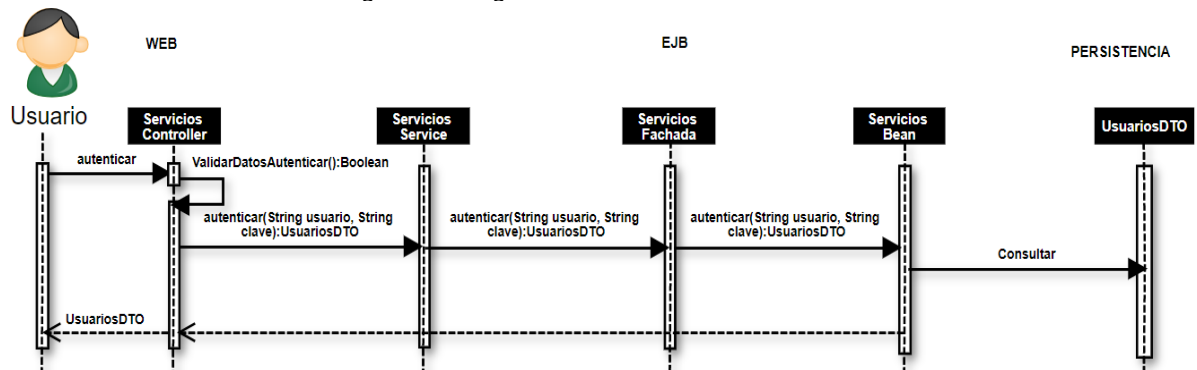


Fuentes. Autores

En la figura 18, se pueden evidenciar las tablas candidatas a conformar, después de realizar el respectivo análisis, y las relaciones de la base de datos se concluye que las tablas Ingreso, opciones, persona, personaOpcion, instituciones y rol son las que mayor relación tienen para conformar el DTO de UsuariosDTO.

1.2 Diagrama de Secuencia: En el diagrama de secuencias se trabajó con el DTO resultado del análisis y la selección de las clases descritas anteriormente, con el fin de realizar el transporte de los objetos que transporta en el proceso, con el fin de facilitar el proceso de autenticación.

Figura 19. Diagrama De Secuencia Autentica

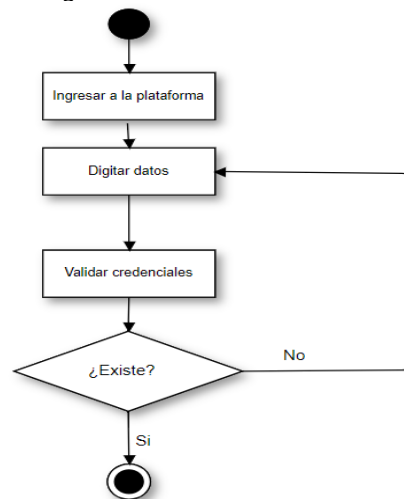


Fuente. Autores

En este primer proceso se realiza la validación de los datos ingreso por el usuario (usuario, contraseña), se envían dos cadenas de texto, pasando del web service al EJB (Enterprise Java Bean) donde están los métodos de transacción que se usarán para el intercambio de información entre aplicación y base de datos de la plataforma. En esta parte se toman los datos de ingreso y se procede a realizar toda la parte de la persistencia, así retornando un objeto de tipo UsuariosDTO.

1.3 Diagrama De Actividades: Para el servicio de autenticar, el usuario primero debe ingresar a la plataforma y digitar el usuario y contraseña, para así poder acceder a la misma.

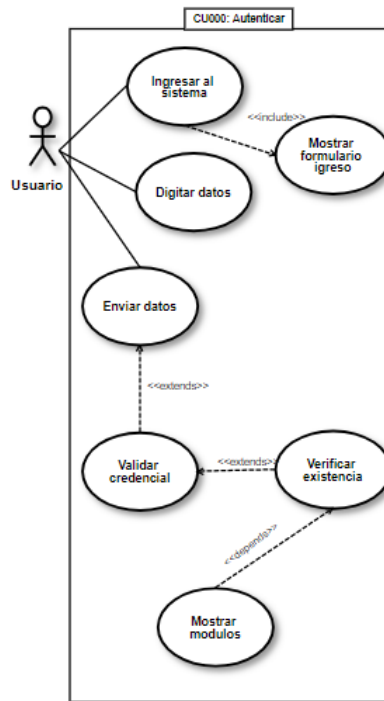
Figura 20. Diagrama De Actividades Autenticar



Fuente. Autores

1.4 Diagrama Casos de Uso: Para este diagrama se puede visualizar las acciones a realizar por el usuario dentro de la plataforma.

Figura 21. Diagrama Casos de Uso Autenticar

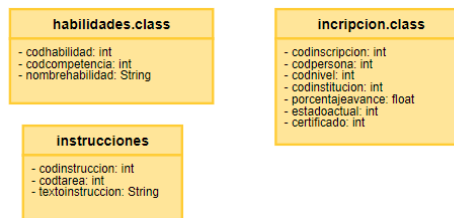


Fuentes. Autores

2. Cursos Permitidos DTO

2.1 Diagrama de Clases: En este servicio se usaron tres clases, las cuales fueron tomadas de la plataforma web que está en construcción, para su debido funcionamiento. Como se puede observar en la siguiente figura:

Figura 22. Diagrama de Clases Cursos Permitidos



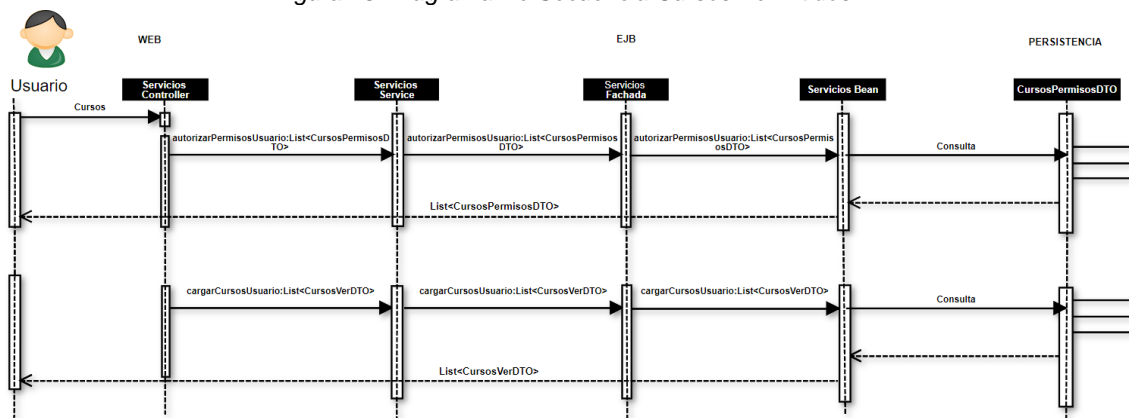
Fuentes. Autores

En la figura 22, se pueden observar las tablas candidatas a conformar, que se eligieron cuando se realizó el respectivo análisis, y las relaciones de la base de datos, las cuales fueron la tabla de habilidades, inscripción e instrucciones; son las que mayor relación tienen para conformar el DTO de CursosPermisosDTO.

2.2 Diagrama de Secuencia

En el diagrama de secuencias cuando el usuario ingresa a la plataforma y se validan los datos devolverá un identificador, el cual se usará para listar los cursos y diferentes temáticas de estos, según los permisos que tenga, retornará una lista de CursosPermitidosDTO y por otra parte CursosVerDTO. Para poder visualizar el diagrama de secuencia completo, remitirse al [Anexo 4. Diagrama de Secuencia Cursos Permitidos](#).

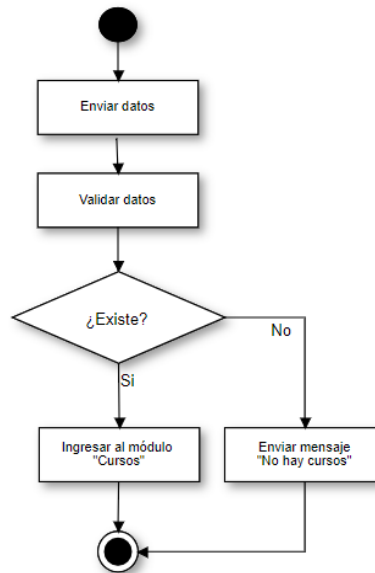
Figura 23. Diagrama De Secuencia Cursos Permitidos



Fuente. Autores

2.3 Diagrama De Actividades: Para el servicio Cursos Permitidos, cuando el usuario ingresa sus datos, el sistema validará la información para así poder mostrar el módulo correspondiente.

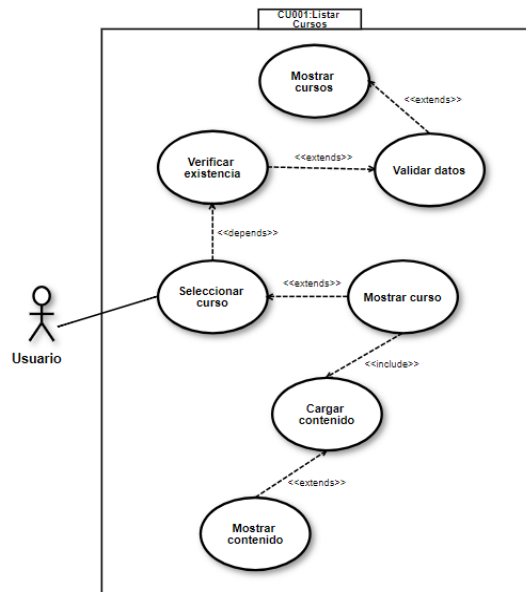
Figura 24. Diagrama De Actividades Cursos Permitidos



Fuente. Autores

2.4 Diagrama Casos de Uso: En el siguiente diagrama se puede visualizar las acciones que puede realizar el usuario dentro de la plataforma.

Figura 25. Diagrama Casos de Uso Cursos Permitidos



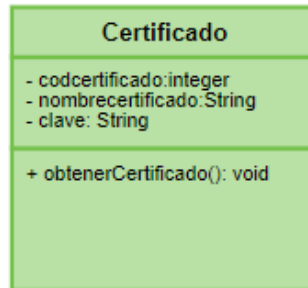
Fuentes. Autores

Es así como se describe la interacción del servicio listar cursos.

3. Obtener certificados

3.1 Diagrama de Clases: En este servicio se usó una clase, la cual fue creada para poder obtener un certificado por cada curso terminado. Como se puede observar en la siguiente figura:

Figura 26. Diagrama de Clases Cursos Permitidos

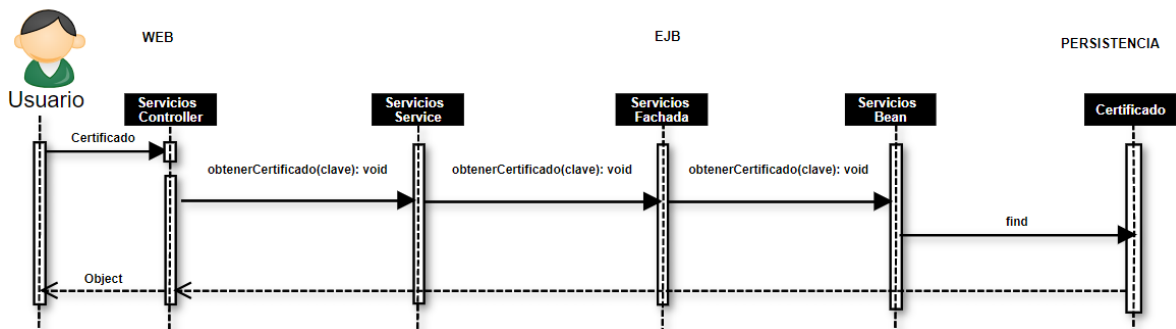


Fuentes. Autores

En la figura 26, se puede ver la tabla candidata, la cual fue necesaria crear ya que no existía, pero que tiene relación con la base de datos de tal forma que se conforme el DTO Obtener Certificado.

3.2 Diagrama de Secuencia: En el diagrama de secuencias, por medio del identificador obtenido anteriormente se podrá realizar el proceso para obtener un certificado por medio de una clave, de tal manera que retorne un objeto.

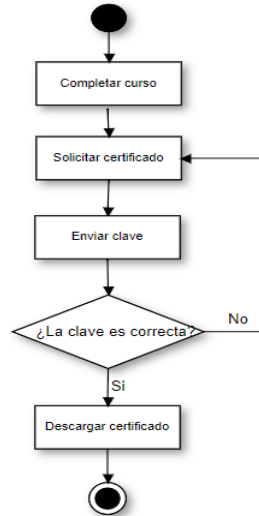
Figura 27. Diagrama de Secuencia Cursos Permitidos



Fuente. Autores

3.3 Diagrama De Actividades: En el servicio Obtener Certificado, el usuario debe ingresar una clave la cual le permita obtenerlo.

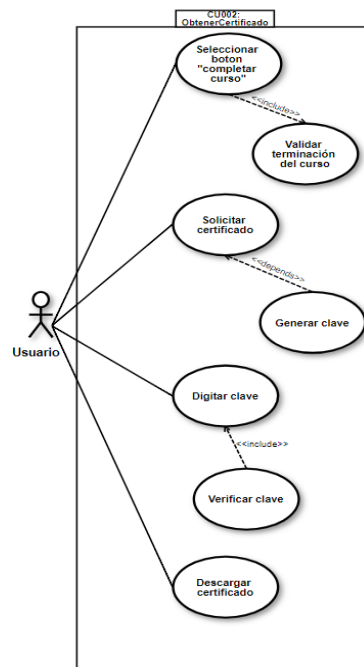
Figura 28. Diagrama De Actividades Cursos Permitidos



Fuente. Autores.

3.4 Diagrama Casos de Uso: En este diagrama se puede visualizar las acciones que puede realizar el usuario dentro de la plataforma para obtener un certificado.

Figura 29. Diagrama Casos de Uso Obtener Certificado



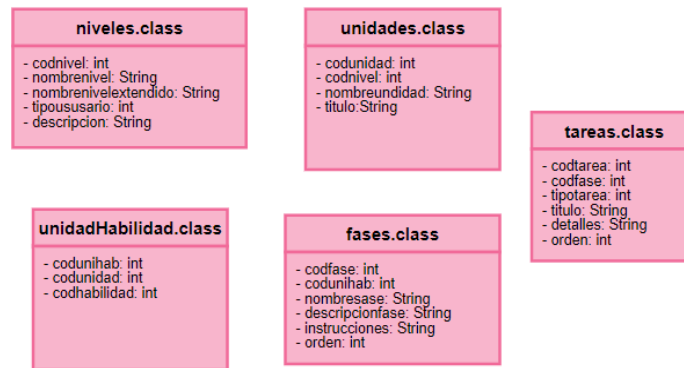
Fuentes. Autores

4. Actualizar curso

4.1 Diagrama de Clases: Para este servicio se usaron cinco clases, las cuales fueron tomadas de la plataforma web que está en construcción, para su debido funcionamiento.

A continuación, se puede observar en la siguiente figura.

Figura 30. Diagrama de Clases Actualizar Curso

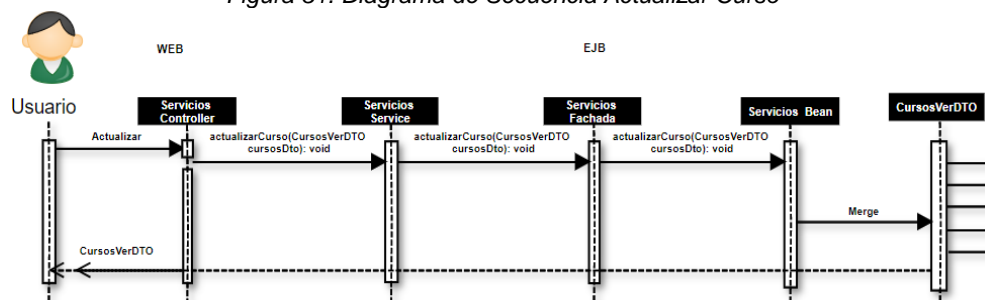


Fuentes. Autores

En la figura 30, se pueden observar las tablas candidatas, después de realizar el respectivo análisis, y las relaciones de la base de datos se concluye que las tablas niveles, unidades, tareas, unidadHabilidad y fases son las que mayor relación tienen para conformar el DTO de CursosVerDTO.

4.2 Diagrama de Secuencia: En el diagrama de secuencias se representa el proceso para actualizar los cursos, y su contenido, por medio del identificador, anteriormente obtenido, de tal manera que retorne un objeto de tipo CursosVerDTO. Para poder visualizar el diagrama de secuencia completo, remitirse al [Anexo 5. Diagrama de Secuencia Actualizar Curso](#).

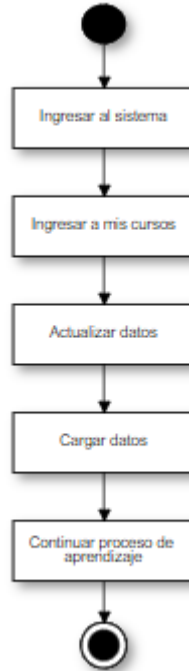
Figura 31. Diagrama de Secuencia Actualizar Curso



Fuente. Autores

4.3 Diagrama De Actividades: En este servicio, el usuario podrá ingresar al módulo y continuar con el proceso de aprendizaje.

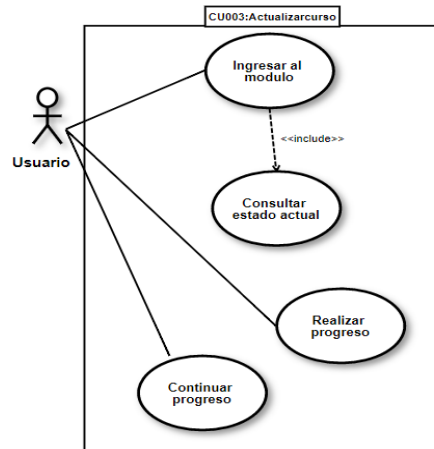
Figura 32. Diagrama De Actividades Actualizar Curso



Fuente. Autores.

4.4 Diagrama Casos de Uso: En este diagrama se puede visualizar las acciones que podrá realizar el usuario dentro de la plataforma para seguir con el proceso de aprendizaje.

Figura 33. Diagrama Casos de Uso Actualizar Curso

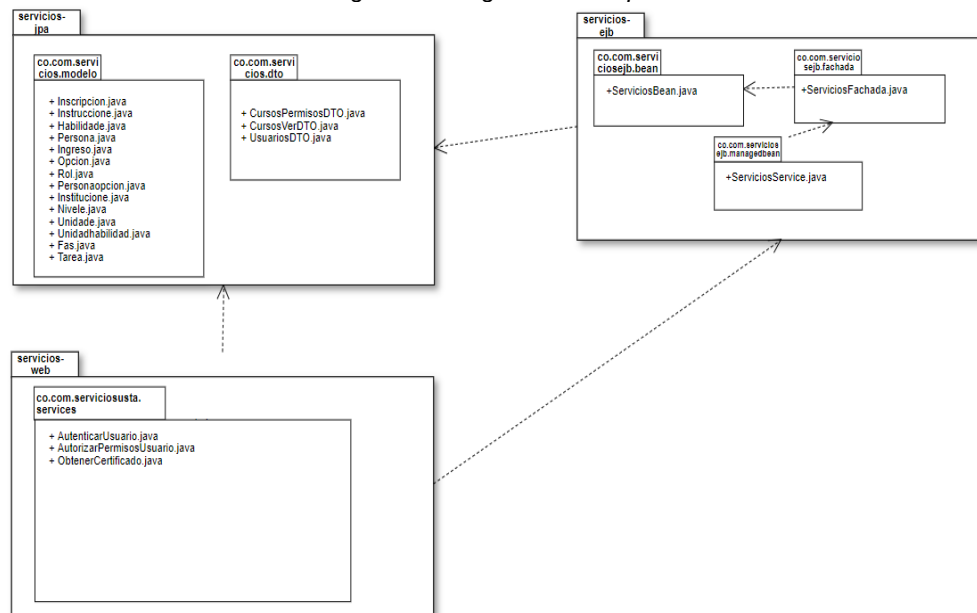


Fuentes. Autores

Diagrama De Paquetes

En el diagrama de paquetes se observa la estructura y agrupación por paquetes que se tiene en el modelo, en el primer paquete se tienen los servicios del JPA, se realizó el mapeo de todas las entidades de la base de datos, estas clases se encuentran en el paquete `co.com.servicios.modelo` con sus respectivos atributos, getter y setter.

Figura 34. Diagrama de Paquetes



Fuentes. Autores

En el paquete `co.com.servicios.dto` tienen unas clases `CursosPermisosDTO.java`, `CursosVerDTO.java`, `UsuariosDTO.java`, que tienen como función crear un objeto de cada uno de estos y así facilitar la extracción de los datos requeridos.

En el paquete de `servicios-ejb` se crean tres paquetes donde se crean todos los métodos que se necesitan para el intercambio de información de aplicación a base de datos y la base de datos a la aplicación. El paquete `co.com.servicioejb.fachada` se encuentra la funcionalidad de todos los métodos que se usarán y consume el paquete `co.com.servicioejb.bean` donde contienen los cascarones de estos métodos y en el paquete `co.com.servicioejb.managedbean` se tiene una clase que es la que consume los servicios fachada para poder encapsular estos métodos y reducir la dificultad.

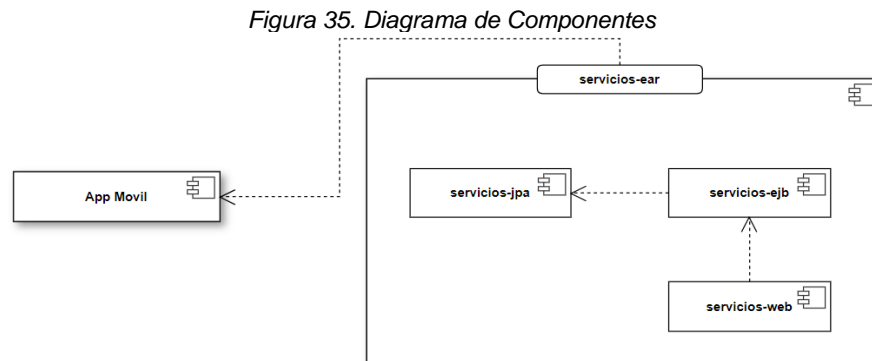
En el paquete de servicios-web se crea un paquete `co.com.serviciosusta.services` donde se crean los servicios los cuales se desplegaran en el servidor para que puedan ser consumidas por la plataforma web.

El paquete de servicios-ejb consume el paquete de servicios-jpa para poder hacer uso de las clases mapeadas para los métodos realizados en esta.

El paquete de servicios web es el que consume el paquete de servicios-jpa donde se tienen todas las clases de las entidades de la base de datos y también consume el paquete de servicios-ejb, se encuentran los métodos creados y hará uso de estos para la implementación de sus servicios.

Diagrama De Componentes

En el diagrama de componentes se puede observar los elementos del diseño para la aplicación. En esta parte podemos observar la estructura de la plataforma y cómo se relacionan entre ellos.



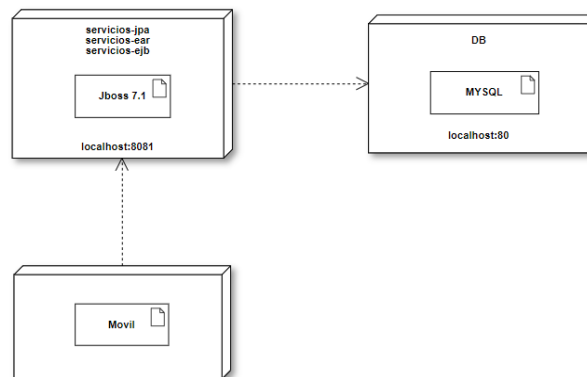
Fuentes. Autores

En este sector se tiene la parte de todo lo que es el front la vista del usuario que será móvil, este segmento consume los servicios-web, que es donde se encuentran todos los servicios que se desplegaran en el servidor para el debido funcionamiento del Front. Los servicios-web consumen los servicios ejb que se encuentran la implementación de los métodos anteriormente creados y finalmente los servicios-ejb es el que consume los servicios-jpa donde se encuentran todas las entidades relacionadas con la base de datos y sus atributos para poder relacionarlos con los métodos anteriormente creados.

Diagrama De Despliegue

El diagrama de despliegue es un diagrama de Lenguaje Unificado de Modelado, el cual permite ver las relaciones físicas que se tienen en los diferentes nodos que componen el sistema y cómo se reparten de los componentes sobre estos.

Figura 36. Diagrama de Despliegue



Fuentes. Autores

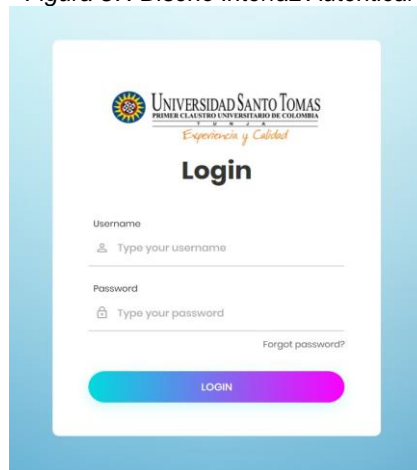
En la Figura 36 se puede observar que servidores se están usando para la implementación de la plataforma, en la parte del front será desplegada en una aplicación móvil, para las interfaces y el consumo de los servicios se realizará por medio de Jboss 7.1 el cual brindará los WDLs que permitirá el acceso a los Web Services. En el servidor Jboss 7.1 se encuentra toda la implementación de métodos que contiene servicios-ejb, el mapeo de las entidades de la base de datos que se encuentra en servicios-jpa y finalmente el servicios-ear que es el que nos permite realizar el despliegue del servidor y contiene el empaquetado de todos los proyectos, de los métodos y entidades utilizados. Finalmente, los servicios-jpa, servicios-ejb y servicios-ear consumen la base de datos proyecto, que se realizó con MySQL donde se tienen todas las tablas, atributos, que usara servicios-jpa para poder realizar el mapeo de las entidades y para que puedan ser utilizadas por los servicios-ejb.

Adicionalmente, del desarrollo de este modelo, se realizaron los diseños de interfaz de usuario de cada uno de los servicios para el catálogo de servicios de la USTA.

A continuación, se podrá evidenciar la realización de los mismos:

1. **Autenticar:** Se sugiere este diseño para el servicio de autenticar, donde el usuario debe ingresar el correo(usuario), contraseña y seleccionar el botón de Login para poder acceder a la plataforma.

Figura 37. Diseño Interfaz Autenticar

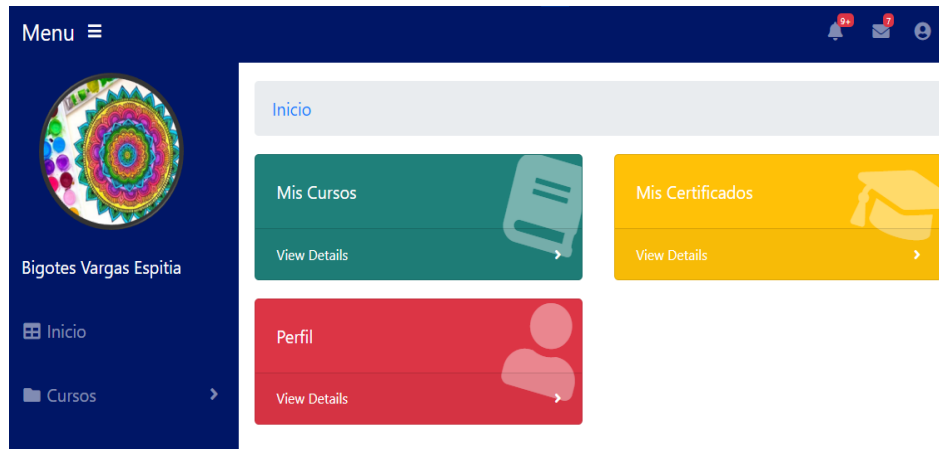
The image shows a login form for Universidad Santo Tomás. At the top, there is the university's logo and name: "UNIVERSIDAD SANTO TOMÁS" with the tagline "PRIMER CLAUSTRO UNIVERSIDAD DE COLOMBIA" and "Experiencia y Calidad". Below this, the word "Login" is centered in a bold font. The form contains two input fields: "Username" with a placeholder "Type your username" and "Password" with a placeholder "Type your password". A "Forgot password?" link is located to the right of the password field. At the bottom of the form is a large, rounded "LOGIN" button with a blue-to-purple gradient.

Fuentes. Autores

2. Autorizar:

Cuando el usuario ingresa a la plataforma, se sugiere este diseño como pantalla principal, la cual tendrá un menú lateral izquierdo donde podrá acceder a los cursos y podrá observar más opciones que le brindará la plataforma cómo ver perfil, mis cursos y mis certificados.

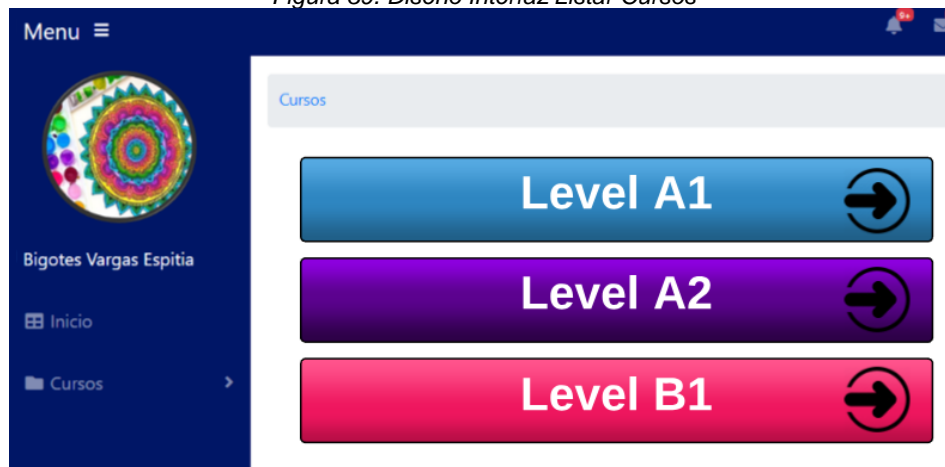
Figura 38. Diseño Interfaz Autorizar



Fuentes. Autores

3. Listar Cursos: Cuando el usuario elige la opción del menú lateral Cursos, se listarán los cursos disponibles por la plataforma.

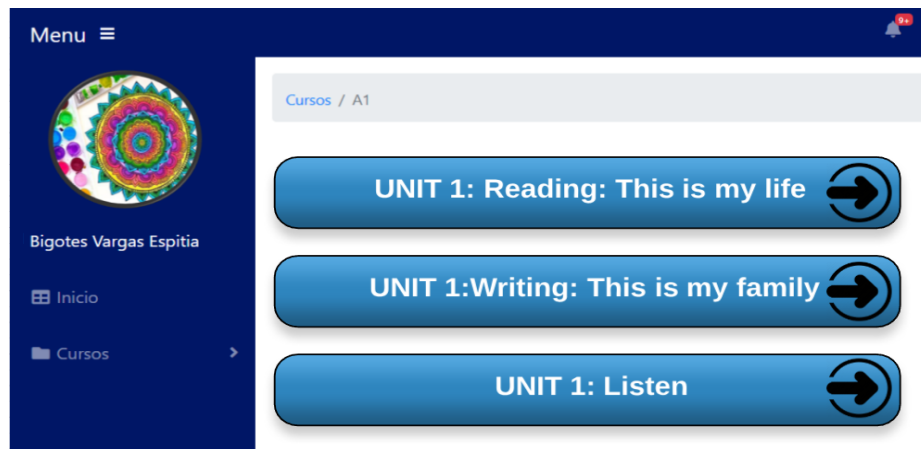
Figura 39. Diseño Interfaz Listar Cursos



Fuentes. Autores

4. Obtener los subtemas de la temática: Al seleccionar el curso deseado, se mostrarán los subtemas pertenecientes a este.

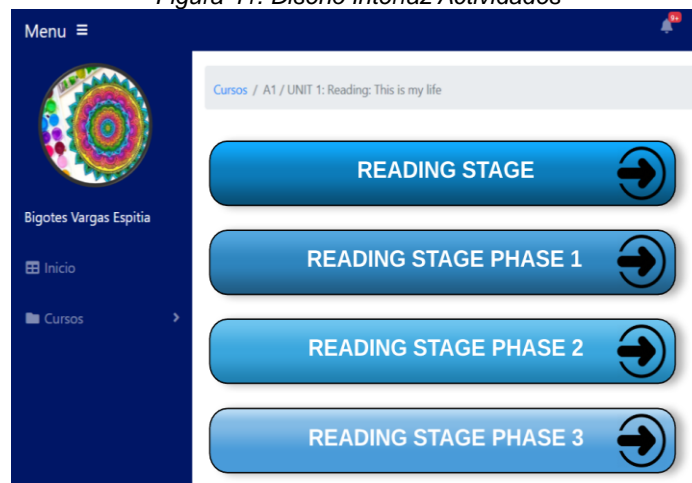
Figura 40. Diseño Interfaz Subtemas



Fuentes. Autores

5. Obtener lista de actividades del subtema elegido: Posteriormente de elegir el subtema, se mostrarán las actividades contenidas por este.

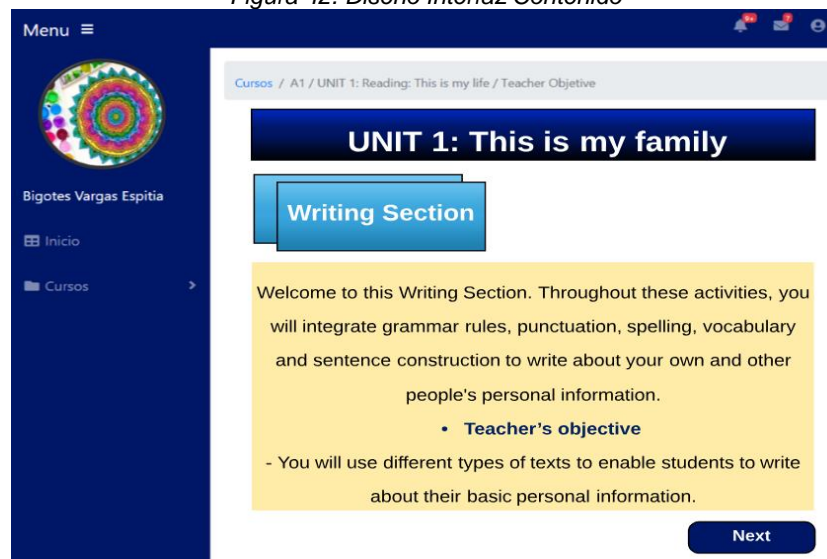
Figura 41. Diseño Interfaz Actividades



Fuentes. Autores

6. Obtener el Contenido de aprendizaje: Finalmente, se podrá visualizar el contenido de aprendizaje de las actividades propuestas por cada curso.

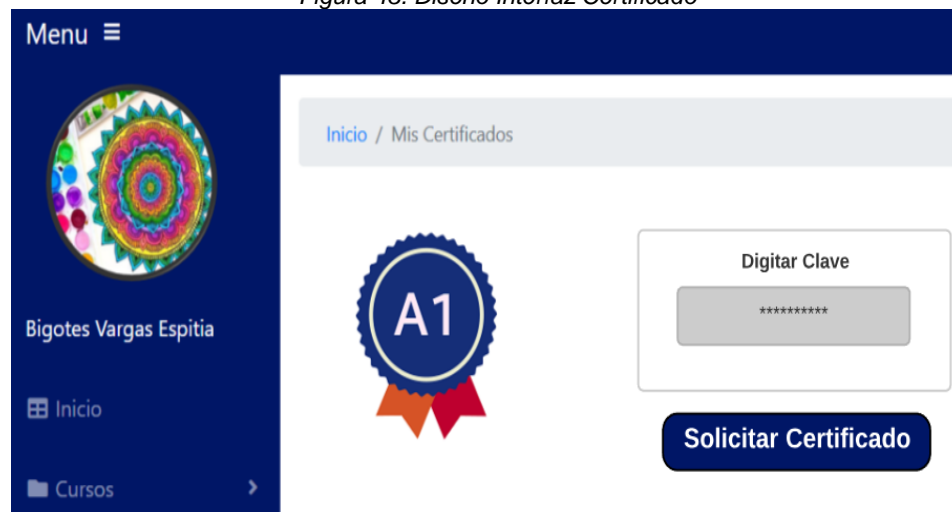
Figura 42. Diseño Interfaz Contenido



Fuentes. Autores

7. Obtener Certificado de aprendizaje: En este servicio el usuario podrá observar sus certificados, que ha obtenido al completar los cursos realizados y solicitar los nuevos.

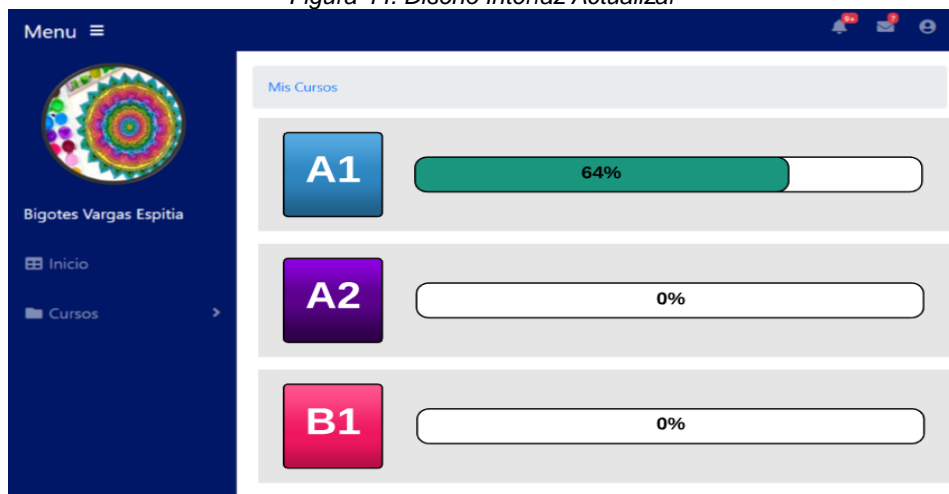
Figura 43. Diseño Interfaz Certificado



Fuentes. Autores

8. Actualizar resultado: Cuando el usuario ingresa a mis cursos, podrá observar el progreso del curso que está realizando y los que ya ha culminado, para que pueda seguir con el aprendizaje.

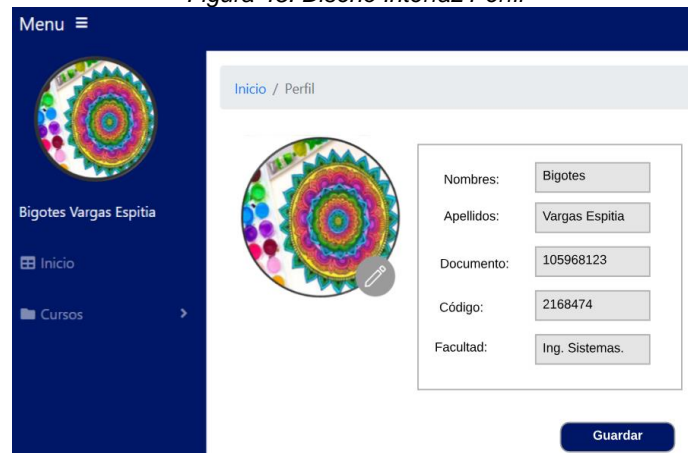
Figura 44. Diseño Interfaz Actualizar



Fuentes. Autores

9. **Mostrar Perfil:** En perfil el usuario podrá observar, editar y actualizar sus datos personales.

Figura 45. Diseño Interfaz Perfil



Fuentes. Autores

6.4 Fase 4. Desarrollo de la aplicación

En esta fase se construyó el maquetado y cascarón con el lenguaje de programación Java y el entorno de desarrollo Eclipse Oxygen, de los servicios anteriormente definidos y se realizaron las debidas pruebas de implementación del mismo.

Tabla 6. Fase 4

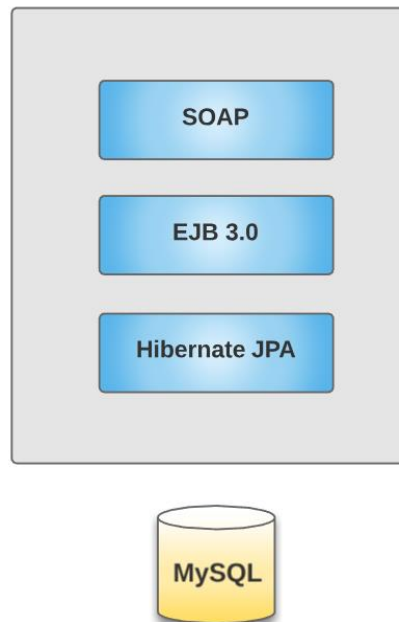
Tipo	Elementos
Entradas	<ul style="list-style-type: none"> ● Diagramas del Modelo 4 + 1 Vistas ● Introspección del código
Instrumentos y/o herramientas	<ul style="list-style-type: none"> ● Sitio de trabajo ● Conexión a internet ● Computador ● Programas del computador <ul style="list-style-type: none"> ✓ Navegador Web ✓ Word ✓ Eclipse ✓ MySQL ✓ XAMPP ✓ Android Studio
Salidas	Maquetado y cascarón de los servicios en Java Enterprise Edition y pruebas de implementación.
Tiempo	Cuatro Semanas

Fuente: Autor

En la codificación se utilizaron las siguientes tecnologías: MAVEN en la versión 3.5.2, es el gestor de dependencias y compilador, JEE, para el ambiente de desarrollo se utiliza la jdk 1.8 y para el servidor de aplicaciones JBOSS versión 7 que utiliza la jdk 1.7 como compilador, JPA como Api de mapeo de las entidades relacionales e HIBERNATE como framework de persistencia y mapeo a base de datos, EJB versión 3.0 para el control de transacciones entre aplicación y base de datos, y SOAP como tecnología para proveer la comunicación de los servicios web.

Con base a esas tecnologías escogidas se propone la siguiente arquitectura:

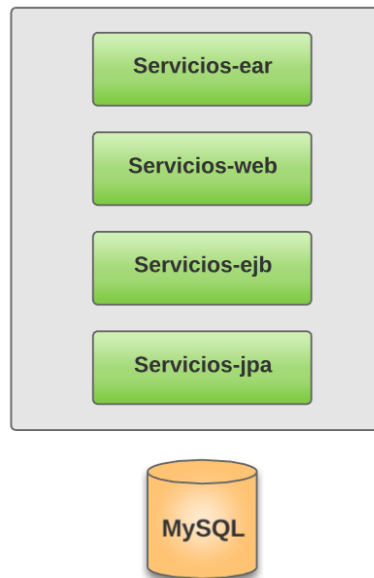
Figura 46. Modelo Arquitectónico 1



Fuentes. Autores

En este modelo del proyecto, se puede observar la estructura que se usó para la construcción del maquetado y sus respectivos cascarones para la plataforma. Esta está conformada por un proyecto java Servicios-ear, el cual es el encargado de empaquetar los módulos del Web, EJB, JPA que incluyen servicios web que se van a exponer, este ejecuta una herramienta de línea de mandos para luego ensamblarlos y así realizar el despliegue para generar el archivo WAR (archivador de aplicación web), en Servicios-web, se crean los servicios que incluyen los métodos necesarios para el transporte de información, el Servicios-ejb, se realiza la transacción de información entre la aplicación y base de datos, también se encuentra Servicios-jpa está toda la parte de persistencia y el mapeo de las entidades y finalmente se encuentra, tal y como se puede detallar en la figura 47, MySQL como motor de la base de datos y Apache como servidor web HTTP.

Figura 47. Modelo del proyecto



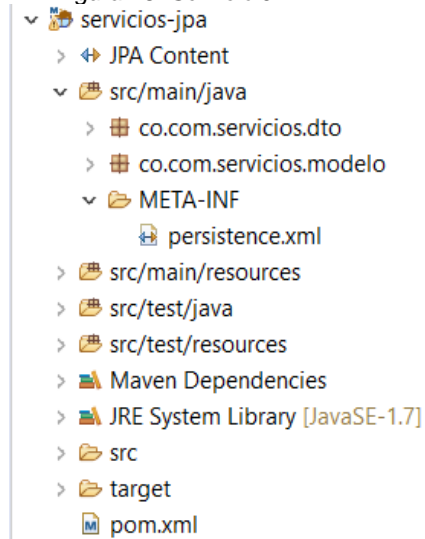
Fuente. Autores

A continuación, se podrá evidenciar la estructura del proyecto creado en eclipse.

Servicios-jpa: Este proyecto está conformado por la siguiente estructura:

- co.com.servicios.modelo: En este paquete se tienen las entidades de la base de datos reflejadas en código java.
- co.com-servicios.dto: En esta parte se tienen los objetos creados a partir de las entidades clasificadas anteriormente, como objetos de transporte de información.

Figura 48. Servicio JPA



INGENIERÍA DE SISTEMAS
INFORME FINAL DESARROLLO TECNOLÓGICO
(plantilla en construcción)

PÁGINA 54
DE 83
VERSIÓN: 01a

Fuentes. Autores

- META-INF: En esta carpeta contiene el archivo persistence.xml, donde se encuentran las entidades de la base de datos, como se puede observar en la siguiente figura:

Figura 49. JPA persistence.xml

```
persistence.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="2.1"
3     xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
5   <persistence-unit name="servicios-jpa">
6     <class>co.com.servicios.modelo.Auditoria</class>
7     <class>co.com.servicios.modelo.Competencia</class>
8     <class>co.com.servicios.modelo.Cuadroautoevaluacion</class>
9     <class>co.com.servicios.modelo.CuadroautoevaluacionPK</class>
10    <class>co.com.servicios.modelo.Evaluacion</class>
11    <class>co.com.servicios.modelo.Fase</class>
12    <class>co.com.servicios.modelo.Habilidad</class>
13    <class>co.com.servicios.modelo.Ingresos</class>
14    <class>co.com.servicios.modelo.Inscripcion</class>
15    <class>co.com.servicios.modelo.Institucion</class>
16    <class>co.com.servicios.modelo.Instruccion</class>
17    <class>co.com.servicios.modelo.Nivele</class>
18    <class>co.com.servicios.modelo.Objetivo</class>
19    <class>co.com.servicios.modelo.Opcion</class>
20    <class>co.com.servicios.modelo.Opcionespreguntaseva</class>
21    <class>co.com.servicios.modelo.Opcionespreguntastarea</class>
22    <class>co.com.servicios.modelo.Persona</class>
23    <class>co.com.servicios.modelo.Preguntaseva</class>
24    <class>co.com.servicios.modelo.Preguntastarea</class>
25    <class>co.com.servicios.modelo.Recursospreta</class>
26    <class>co.com.servicios.modelo.Reportetarea</class>
27    <class>co.com.servicios.modelo.ReportetareaPK</class>
28    <class>co.com.servicios.modelo.Respuestapreta</class>
29    <class>co.com.servicios.modelo.RespuestapretaPK</class>
30    <class>co.com.servicios.modelo.Rol</class>
31    <class>co.com.servicios.modelo.Tarearecurso</class>
32    <class>co.com.servicios.modelo.TarearecursoPK</class>
33    <class>co.com.servicios.modelo.Tarea</class>
34    <class>co.com.servicios.modelo.Unidad</class>
35    <class>co.com.servicios.modelo.Unidadhabilidad</class>
36    <class>co.com.servicios.modelo.Personaopcion</class>
37    <class>co.com.servicios.modelo.PersonaopcionPK</class>
```

Fuentes. Autores

En el archivo pom.xml, es un fichero XML, la unidad principal del proyecto Maven, el cual contiene información acerca del proyecto, librerías que lo conforman, fuentes, test, dependencias, plugins, versión, entre otros. A continuación, se puede ver su estructura.

Figura 50. JPA pom.xml

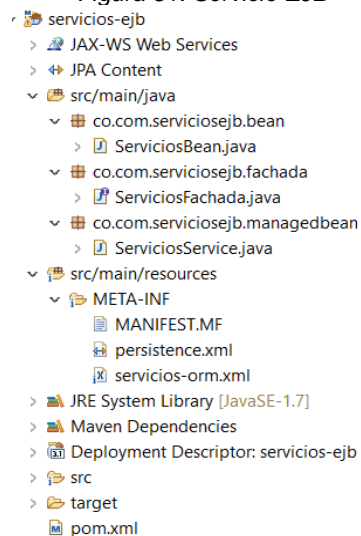
```
*servicios-jpa/pom.xml
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>co.com.servicios</groupId>
5   <artifactId>servicios-jpa</artifactId>
6   <version>0.0.1-SNAPSHOT</version>
7
8   <build>
9     <plugins>
10      <plugin>
11        <groupId>org.apache.maven.plugins</groupId>
12        <artifactId>maven-compiler-plugin</artifactId>
13        <version>2.3.2</version>
14        <configuration>
15          <source>1.7</source>
16          <target>1.7</target>
17        </configuration>
18      </plugin>
19    </plugins>
20  </build>
21
22  <repositories>
23    <repository>
24      <id>Java.Net</id>
25      <url>http://download.java.net/maven/2/</url>
26    </repository>
27  </repositories>
28
29  <dependencies>
30    <dependency>
31      <groupId>javax</groupId>
32      <artifactId>javasee-api</artifactId>
33      <version>6.0</version>
34      <scope>provided</scope>
35    </dependency>
36    <dependency>
37      <groupId>javax.validation</groupId>
```

Fuentes. Autores

servicios-ejb: Este proyecto está conformado por la siguiente estructura: se encarga de realizar el llamado a la fachada para enviar la petición obtenida del proyecto web.

- co.com.serviciosejb.bean: Esta capa se encuentran las clases encargadas de la realizar las llamadas a las transacciones entre el proyecto java y el motor de base de datos.
- co.com.serviciosejb.fachada: Se ubican las interfaces que hacen parte del patrón facade encargado del llamado entre la capa servicio y el bean de transacción.
- co.com.serviciosejb.managedbean: Esta capa se encuentran las clases encargadas de realizar las llamadas a las transacciones entre el proyecto java y el motor de base de datos.

Figura 51. Servicio EJB



Fuentes. Autores

META-INF: En esta carpeta se tiene un archivo persistence.xml, donde se encuentran las entidades de la base de datos, como se puede observar en la siguiente figura:

Figura 52. EJB persistence.xml

```
*persistence.xml ☒
1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="2.1"
3     xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema
4     xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns,
5     <persistence-unit name="superint-dbsuper-jpa">
6         <provider>org.hibernate.ejb.HibernatePersistence</provider>
7         <jta-data-source>java:jboss/env/jdbc/serviciosWs</jta-data-source>
8         <mapping-file>META-INF/servicios-orm.xml</mapping-file>
9         <class>co.com.servicios.modelo.Auditoria</class>
10        <class>co.com.servicios.modelo.Competencia</class>
11        <class>co.com.servicios.modelo.Cuadroautoevaluacion</class>
12        <class>co.com.servicios.modelo.CuadroautoevaluacionPK</class>
13        <class>co.com.servicios.modelo.Evaluacion</class>
14        <class>co.com.servicios.modelo.Fas</class>
15        <class>co.com.servicios.modelo.Habilidad</class>
```

Fuentes. Autores

En los servicios-orm.xml, es el archivo donde se realizan las consultas que se necesiten en los métodos creados.

Figura 53. EJB servicios-orm.xml

```
servicios-orm.xml ☒
1 <?xml version="1.0" encoding="UTF-8"?>
2 <entity-mappings version="2.0"
3     xmlns="http://java.sun.com/xml/ns/persistence/orm"
4     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5     xsi:schemaLocation="http://java.sun.com/xml/ns/persistence/orm http://java.sun.com/xml/ns/persistence/orm_2_0.xsd">
6
7     <named-query name="Persona.CONSULTAR_USUARIO">
8         <query>
9             select p
10            from Persona p
11            where p.usuario = :usuario
12            and p.clave = :clave
13        </query>
14    </named-query>
15
16    <named-native-query name="Persona.CONSULTAR_USUARIO1" result-class="co.com.servicios.modelo.Persona">
17        <query>
18            select p.*
19            from persona p
20            where p.usuario = :usuario
21            and p.clave = :clave
22        </query>
23    </named-native-query>
24
25 </entity-mappings>
```

Fuentes. Autores

En el archivo pom.xml, es un fichero XML, la unidad principal del proyecto Maven, el cual contiene información acerca del proyecto, fuentes, test, dependencias, plugins, versión, entre otros. A continuación, se puede ver su estructura.

Figura 54. EJB pom.xml

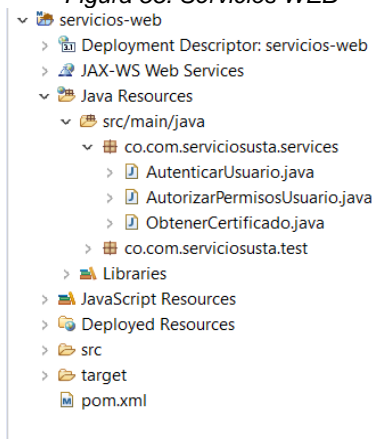
```
servicios-ejb/pom.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
3   <modelVersion>4.0.0</modelVersion>
4
5   <groupId>co.com.servicios</groupId>
6   <artifactId>servicios-ejb</artifactId>
7   <version>0.0.1-SNAPSHOT</version>
8   <packaging>ejb</packaging>
9
10  <name>servicios-ejb</name>
11
12  <properties>
13    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
14  </properties>
15
16  <dependencies>
17    <dependency>
18      <groupId>javax</groupId>
19      <artifactId>javaee-api</artifactId>
20      <version>7.0</version>
21      <scope>provided</scope>
22    </dependency>
23    <dependency>
24      <groupId>mysql</groupId>
25      <artifactId>mysql-connector-java</artifactId>
26      <version>5.1.10</version>
27    </dependency>
28    <dependency>
29      <groupId>javax.persistence</groupId>
30      <artifactId>persistence-api</artifactId>
31      <version>1.0</version>
32      <scope>provided</scope>
33    </dependency>
```

Fuente. Autores.

Servicios-web Este servicio está conformado por la siguiente estructura:

- co.com.serviciosusta.services: En este proyecto se generan los servicios web que se expondrán y serán consumidos con los métodos clasificados según servicio creado.

Figura 55. Servicios WEB



Fuentes. Autores

En el archivo pom.xml, es un fichero XML, la unidad principal del proyecto Maven, el cual contiene información relacionada con librerías que utilizan el proyecto, fuentes, test, dependencias, plugins, versión, entre otros. A continuación, se puede ver su estructura.

Figura 56. WEB pom.xml

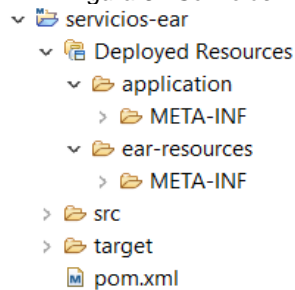
```
servicios-web/pom.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema
3   <modelVersion>4.0.0</modelVersion>
4
5   <groupId>co.com.servicios</groupId>
6   <artifactId>servicios-web</artifactId>
7   <version>0.0.1-SNAPSHOT</version>
8   <packaging>war</packaging>
9
10  <name>servicios-web</name>
11
12  <properties>
13    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
14  </properties>
15
16  <repositories>
17    <repository>
18      <id>maven2-repository.jboss.com</id>
19      <name>JBoss Repository for Maven</name>
20      <url>http://repository.jboss.com/maven2</url>
21    </repository>
22    <repository>
23      <id>prime-repo</id>
24      <name>Prime Technology Maven Repository</name>
25      <url>http://repository.primefaces.org</url>
26      <layout>default</layout>
27    </repository>
28    <repository>
29      <id>maven2-repository.dev.java.net</id>
30      <name>Java.net Repository for Maven</name>
31      <url>http://download.java.net/maven/2/</url>
32      <layout>default</layout>
33    </repository>
34  </repositories>
35
```

Fuentes. Autores

Servicios-ear:

Este proyecto es el encargado de desplegar en el servidor, debido a que contiene los proyectos anteriormente descritos.

Figura 57. Servicios EAR



Fuentes. Autores

En esta parte se puede observar cómo se empaquetan los diferentes proyectos, para realizar el despliegue correspondiente.

Figura 58. EAR pom.xml

```

<modules>
  <jarModule>
    <groupId>co.com.servicios</groupId>
    <artifactId>servicios-jpa</artifactId>
  </jarModule>
  <ejbModule>
    <groupId>co.com.servicios</groupId>
    <artifactId>servicios-ejb</artifactId>
  </ejbModule>
  <webModule>
    <groupId>co.com.servicios</groupId>
    <artifactId>servicios-web</artifactId>
  </webModule>
</modules>
    
```

Fuentes. Autores

6.5. Fase 5. Despliegue del Catálogo

Para el desarrollo de esta fase, fue necesario ensamblar los cascarones anteriormente realizados, en el maquetado, el cual fue importante para poder realizar el despliegue del catálogo.

Tabla 7. Fase 5

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> ● Cascarones y maquetado ● Modelo 4 + 1 vistas
Instrumentos y/o herramientas	<ul style="list-style-type: none"> ● Sitio de trabajo ● Conexión a internet ● Computador ● Programas del computador ✓ Navegador Web ✓ Word ✓ Eclipse ✓ Xampp ✓ MySQL ✓ SoapUI ✓ Android Studio
Salidas	Despliegue de la plataforma.
Tiempo	Cinco Semanas

Fuente: Autores

Cascarones y Maquetado: En el proyecto creado en eclipse se crearon los cascarones y maquetado, en cada uno de los componentes y estructura anteriormente definida.

Como primer paso se crearon los cascarones en el paquete de servicios-ejb.

En `co.com.servicioejb.bean`: Se crearon las siguientes clases, las cuales se encargan de realizar las transacciones entre el proyecto java y el motor de base de datos.

El primer método que se creó fue el del servicio de autenticar, donde se comparan las dos cadenas de texto ingresada por el usuario, verificando si la existencia en la base de datos, si son correcto se traerán los datos del mismo, los cuales serán usados por los demás servicios.

Figura 59. ServiciosBean Autenticar

```
@LocalBean
public class ServiciosBean implements ServiciosFachada{

    @PersistenceContext(unitName="superint-dbsuper-jpa")
    private EntityManager em;

    /*
     * (non-Javadoc)
     * @see co.com.servicioejb.fachada.ServiciosFachada#autenticar(java.lang.String, java.lang.String)
     */
    @SuppressWarnings("unchecked")
    @Override
    public UsuariosDTO autenticar(String usuario, String clave) throws Exception {
        UsuariosDTO usuarioDto = new UsuariosDTO();
        List<Object> ob = new ArrayList<Object>();

        Query query = em.createNamedQuery(Persona.CONULTAR_USUARIO);

        query.setParameter("usuario", usuario);
        query.setParameter("clave",clave);
        try {
            ob = query.getResultList();
            if(ob.size()>0) {
                for(Object o: ob) {
                    usuarioDto = new UsuariosDTO();
                    Persona persona = new Persona();
                    persona = (Persona) o;
                    usuarioDto.setPersona(persona);
                }
            }
        } catch (NoResultException e) {
            e.printStackTrace();
            usuarioDto = null;
        }

        return usuarioDto;
    }
}
```

Fuentes. Autores

A continuación, se podrá observar los cascarones de los servicios definidos en la fase 3 Diseño y refinación de requerimientos.

Figura 60. ServiciosBean Cascarones

```
/*
 * (non-Javadoc)
 * @see co.com.serviciosejb.fachada.ServiciosFachada#listarCursosUsuario(int)
 */
@Override
public List<CursosPermisosDTO> listarCursosUsuario(int id) throws Exception {
    // TODO Auto-generated method stub
    return null;
}

/*
 * (non-Javadoc)
 * @see co.com.serviciosejb.fachada.ServiciosFachada#obtenerCertificado(java.lang.String)
 */
@Override
public boolean obtenerCertificado(String clave) throws Exception {
    // TODO Auto-generated method stub
    return true;
}

/*
 * (non-Javadoc)
 * @see co.com.serviciosejb.fachada.ServiciosFachada#actualizarCurso(co.com.servicios.dto.CursosVerDTO)
 */
@Override
public void actualizarCurso(CursosVerDTO cursosDto) throws Exception {
    // TODO Auto-generated method stub
}

/*
 * (non-Javadoc)
 * @see co.com.serviciosejb.fachada.ServiciosFachada#cargarCursosUsuario(java.lang.String)
 */
@Override
public List<CursosVerDTO> cargarCursosUsuario(String idCurso) throws Exception {
    // TODO Auto-generated method stub
    return null;
}
```

Fuentes. Autores

En “*co.com.serviciosejb.fachada*”, se crearon los cascarones de los métodos que se usarán, los cuales heredan hacia la clase lógica. Como se puede ver en la siguiente figura:

Figura 61. ServiciosFachada Cascarones

```
package co.com.serviciosejb.fachada;

import java.util.List;

/* Es la capa intermedia de donde heredan los metodos hacia la clase logica, se definen todos los metodos que se van a utilizar.
El EJB de fachada es el que proporciona todos los metodos de establecimiento de datos para los objetos de valor y datos. */

@Remote
public interface ServiciosFachada {
    /**
     * Este método sirve para autenticar el usuario que ingrese a la plataforma
     * @param usuario: String
     * @param clave: String
     * @return UsuarioDTO: retorna objetos de tipo person, Ingreso, Opcion, Rol, Personaopcion e Institucion
     * @throws Exception
     */
    public UsuariosDTO autenticar(String usuario, String clave) throws Exception;

    /**
     * Este método sirve para listar los cursos que ofrece la plataforma al usuario que ingreso a esta.
     * @param id: int
     * @return CursosPermisosDTO: retornara una lista de inscripciones, instrucciones y habilidades;
     * @throws Exception
     */
    public List<CursosPermisosDTO>listarCursosUsuario(int id) throws Exception;

    /**
     * Este método es el que mostrara los certificados que el usuario ha obtenido por cada curso terminado y aprobado
     * @param clave: String
     * @return boolean true si el certificado esta en estado 1: Disponible en la tabla inscripción, false
     * si el certificado esta en estado 2: No disponible en la tabla inscripción.
     * @throws Exception
     */
    public boolean obtenerCertificado(String clave) throws Exception;

    /**
     * Este método le mostrara al usuario los avances de todos los cursos, subtemas y actividades que ha realizado.
     * @param cursosDto: CursosVerDTO
     * @throws Exception
     */
    public void actualizarCurso(CursosVerDTO cursosDto) throws Exception;

    /**
     * Se obtienen los cursos (niveles) al cual el usuario esta inscrito y activo, se obtiene el dato por el
     * metodo de autorizarPermisoUsuario
     * @param cursosDto: CursosVerDTO
     * @throws Exception
     */
    public List<CursosVerDTO> cargarCursosUsuario(String idCurso) throws Exception;
}
```

Fuentes. Autores

En “*co.com.serviciosejb.managedbean*” se crearon los cascarones de los métodos que se necesitan para el intercambio de información de la aplicación con la base de datos y de la base de datos a la aplicación.

Figura 62. Serviciosmanagedbean Cascarones

```
package co.com.serviciosejb.managedbean;

import java.util.List;

/** fachada entre la presentacion y la logica del negocio. En esta parte se crean todos los métodos que se necesitan para el intercambio de
 * información de aplicación a base de datos y la base de datos a la aplicación*/
@Stateless
@LocalBean
@Transactional(TransactionalType.NOT_SUPPORTED)
public class ServiciosService {

    @EJB
    ServiciosFachada fachada;

    /**
     * Este método sirve para autenticar el usuario que ingrese a la plataforma
     * @param usuario: String
     * @param clave: String
     * @return UsuarioDTO: retorna objetos de tipo person, Ingreso, Opcion, Rol, Personaopcion e Instituciones
     * @throws Exception
     */
    public UsuariosDTO autenticar(String usuario, String clave) throws Exception{
        return fachada.autenticar(usuario, clave);
    }

    /**
     * Este método sirve para listar los cursos que ofrece la plataforma al usuario que ingreso a esta.
     * @param id: int
     * @return CursosPermisosDTO: retornara una lista de inscripciones, instrucciones y habilidades;
     * @throws Exception
     */
    public List<CursosPermisosDTO>listarCursosUsuario(int id) throws Exception{
        return null;
    }

    /**
     * Este método es el que mostrara los certificados que el usuario ha obtenido por cada curso terminado y aprobado
     * @param clave: String
     * @return boolean true si el certificado esta en estado 1: Disponible en la tabla inscripción, false
     * si el certificado esta en estado 2: No disponible en la tabla inscripción.
     * @throws Exception
     */
    public boolean obtenerCertificado(String clave) throws Exception{
        return fachada.obtenerCertificado(clave);
    }

    /**
     * Este método le mostrara al usuario los avances de todos los cursos, subtemas y actividades que ha realizado.
     * @param cursosDto: CursosVerDTO
     * @throws Exception
     */
    public void actualizarCurso(CursosVerDTO cursosDto) throws Exception{}

    /**
     * Se obtienen los cursos (niveles) al cual el usuario esta inscrito y activo, se obtiene el dato por el
     * metodo de autorizarPermisoUsuario
     * @param cursosDto: CursosVerDTO
     * @throws Exception
     */
    public List<CursosVerDTO> cargarCursosUsuario(String idCurso) throws Exception{
        List<CursosVerDTO> cursosVerDTOS = null;
        return cursosVerDTOS;
    }
}
```

Fuentes. Autores

En los servicios-web se encuentra “co.com.serviciosusta.services” donde se crearon los servicios web que se expondrán y serán consumidos con los métodos correspondientes.

En el servicio “*AutenticarUsuario*” se comparan las dos cadenas de texto ingresadas por el usuario y según el caso envía un mensaje si no está registrado, de lo contrario devolverá los datos del usuario autenticado.

Figura 63. Servicio-web Autenticar

```
package co.com.serviciosusta.services;

import java.io.Serializable;

@WebService
@Stateless
public class AutenticarUsuario implements Serializable{

    private static final long serialVersionUID = 1L;

    @EJB
    ServiciosService service;

    /**
     * En este metodo se verifican las dos cadenas de texto ingresadas por le usuario para poder realizar la autenticacion
     * @param usuario: String
     * @param contrasena: String
     * @return usuariosDTO
     */

    @WebMethod(operationName="autenticar")
    public UsuariosDTO autenticar(@WebParam(name="usuario") String usuario, @WebParam(name="contrasena") String contrasena){
        UsuariosDTO usuariosDTO = new UsuariosDTO();
        try {
            if (usuario!=null && !usuario.equals("") && contrasena!=null && !contrasena.equals("")) {
                usuariosDTO = service.autenticar(usuario.trim(), contrasena.trim());
                if(usuariosDTO.getPersona().getDocumento()==null || usuariosDTO.getPersona().getDocumento().equals("")) {
                    usuariosDTO.setMensaje("El Usuario no esta registrado");
                }
            }
        } catch (Exception e) {
            usuariosDTO.setMensaje("Error al consultar el usuario");
            Logger.getLogger("Error al consultar el usuario");
        }
        return usuariosDTO;
    }
}
```

Fuentes. Autores

En “*AutorizarPermisosUsuario*” se crearon los cascarones de los métodos necesarios para los permisos establecidos para cada usuario y la carga de cada una de las unidades y sus componentes.

Figura 64. Servicio-web Autorizar

```
package co.com.serviciosusta.services;

import java.util.ArrayList;

@WebService
@Stateless
public class AutorizarPermisosUsuario {

    @EJB
    ServiciosService service;

    /**
     * en este metodo consultamos la inscripcion de usuario ingresando en documento que se obtiene del servicio de autenticar,
     * y retorna la inscripcion para consultar los cursos que esta activo
     * @param idUsuario: String
     * @return verDTOS
     */
    @WebMethod(operationName="autorizarPermisosUsuario")
    public List<CursosPermisosDTO> autorizarPermisosUsuario(@WebParam(name="idUsuario") String idUsuario){
        List<CursosPermisosDTO> verDTOS = new ArrayList<CursosPermisosDTO>();
        try {
            if(idUsuario!=null && !idUsuario.equals("")) {
                verDTOS = service.listarCursosUsuario(Integer.valueOf(idUsuario));
            }
        } catch (Exception e) {
            Logger.getLogger("Error al consultar la inscripcion");
        }
        return verDTOS;
    }

    /**
     * Se obtienen los cursos (niveles) al cual el usuario esta inscrito y activo, se obtiene el dato por el
     * metodo de autorizarPermisoUsuario
     * @param idCurso: String
     * @return verDTOS
     */
    @WebMethod(operationName="cargarCursosUsuario")
    public List<CursosVerDTO> cargarCursosUsuario(@WebParam(name="idCurso") String idCurso){
        List<CursosVerDTO> verDTOS = new ArrayList<CursosVerDTO>();
        try {
            if(idCurso!=null && !idCurso.equals("")) {
                verDTOS = service.cargarCursosUsuario(idCurso);
            }
        } catch (Exception e) {
            Logger.getLogger("Error al cargar el curso");
        }
        return verDTOS;
    }
}
```

Fuentes. Autores

En “*ObtenerCertificado*” se creó el cascarón del método, el cual validará la clave ingresada por el usuario, si la clave es correcta devolverá un objeto con el certificado solicitado, de lo contrario mostrará un mensaje de error.

Figura 65. Servicio-web Obtener certificado

```
package co.com.serviciosusta.services;

import java.util.logging.Logger;

@WebService
@Stateless
public class ObtenerCertificado {

    @EJB
    ServiciosService service;

    /**
     * En este metodo verifica la clave ingresada por el usuario para poder generar el certificado solicitado
     * @param clave
     * @return object
     */
    @WebMethod(operationName="obtenerCertificado")
    public void obtenerCertificado(@WebParam(name="clave") String clave){
        try {
            if(clave!=null && !clave.equals("")) {
                service.obtenerCertificado(clave);
            }
        } catch (Exception e) {
            Logger.getLogger("Error al consultar el inscripcion");
        }
    }
}
```

Fuentes. Autores

A continuación, se observa cómo se despliega en el servidor JBoss, el archivo servicios-ear-0.0.1-SNAPSHOT.ear, el cual está ubicado en el paquete "sevicios-ear", donde se encuentran los proyectos anteriormente descritos.

Figura 66. Servicios-ear despliegue

Deployments

Name	Runtime Name	Enabled	En/Disable	Remove
servicios-ear-0.0.1-SNAPSHOT.ear	servicios-ear-0.0.1-SNAPSHOT.ear	✓	<input type="button" value="Disable"/>	<input type="button" value="Remove"/>

Fuentes. Autores

Después de desplegar el anterior archivo en el servidor de JBoss, nos muestra los servicios web anteriormente expuestos.

Figura 67. Servicio-web Obtener certificado

Web Service Endpoints

Web Service Endpoints. Endpoints need to be deployed as regular applications.

Available Web Service Endpoints

Name	Context	Deployment
AutenticarUsuario	servicios-web	servicios-ear-0.0.1-SNAPSHOT.ear
ObtenerCertificado	servicios-web	servicios-ear-0.0.1-SNAPSHOT.ear
AutorizarPermisosUsuario	servicios-web	servicios-ear-0.0.1-SNAPSHOT.ear

1-3 of 3

Selection

Name: AutenticarUsuario	Context: servicios-web
Class: co.com.serviciosusta.services.AutenticarI	Type: JAXWS_EJB3
WSDL Url: http://192.168.1.124:8081/servicios-web/AutenticarUsuario?wsdl	Deployment: servicios-ear-0.0.1-SNAPSHOT.ear

Fuentes. Autores

Para comprobar la implementación anteriormente realizada, se utilizó el servicio de "AutenticarUsuario" para la verificación del correcto funcionamiento, el cual se realizó con la herramienta SoapUI, para el uso de esta herramienta ingresamos el WSDL de este servicio, el cual nos mostró el XML correspondiente, donde ingresamos el usuario y contraseña de un usuario existente en la base de datos, a continuación, se puede observar en la siguiente figura.

Figura 68. XML Entrada

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:autenticar>
      <!--Optional:-->
      <usuario>decsistemas@ustatunja.edu.co</usuario>
      <!--Optional:-->
      <contrasena>Rq2eqzqFBjK.6GRx8SwLz1FsZIep5K.</contrasena>
    </ser:autenticar>
  </soapenv:Body>
</soapenv:Envelope>
```

Fuentes. Autores

Al ingresar los anteriores datos, se procedió a ejecutar y así poder comprobar su funcionamiento, obteniendo como resultado todos los datos correspondientes al usuario autenticado de tal forma comprobar que el servicio si fue exitoso.

Figura 69. XML Salida

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:autenticarResponse xmlns:ns2="http://services.serviciosusta.com.co/">
      <return>
        <persona>
          <apellidos>Gutierrez LOpez</apellidos>
          <clave>Rq2eqzqFBjK.6GRx8SwLz1FsZIep5K.</clave>
          <codinstitucion>5</codinstitucion>
          <codpersona>3</codpersona>
          <codrol>3</codrol>
          <correo>decsistemas@ustatunja.edu.co</correo>
          <direccion>No tiene direccion</direccion>
          <documento>64582727</documento>
          <fechaexpediciondocumento>1979-01-25T00:00:00-05:00</fechaexpediciondocumento>
          <genero>2</genero>
          <nombres>Luz Elena</nombres>
          <telefono>No tiene cel</telefono>
          <tipodocumento>CC</tipodocumento>
          <usuario>decsistemas@ustatunja.edu.co</usuario>
        </persona>
      </return>
    </ns2:autenticarResponse>
  </soap:Body>
</soap:Envelope>
```

Fuentes. Autores.

Después de evidenciar el funcionamiento de este servicio, se elaboró una aplicación móvil para la implementación del servicio, de tal manera comprobar que fuera exitosa la comunicación entre estas tecnologías.

Para conectar la aplicación móvil con los servicios anteriormente creados se deben establecer cuatro parámetros, a los cuales se le asignará información del WSDL generado en el servidor de Jboss donde se subió el archivo ear, nombre del servicio y demás propiedades como se puede visualizar en la siguiente figura.

Figura 70. Conexión android a servicio Autenticar

```
@RequiresApi(api = Build.VERSION_CODES.CUPCAKE)
private class TareaUser extends AsyncTask<Void, Void, Boolean> {
    @Override
    protected Boolean doInBackground(Void... params) {
        // TODO: attempt authentication against a network service.
        //WebService - Opciones
        final String SOAP_ACTION = "http://192.168.1.124:8081/servicios-web/AutenticarUsuario";
        final String NAMESPACE = "http://services.serviciosusta.com.co/";
        final String METHOD_NAME = "autenticar";
        final String URL="http://192.168.1.124:8081/servicios-web/AutenticarUsuario?wsdl";
    }
}
```

Fuentes. Autores.

Luego de realizar la conexión con los servicios, se toman los datos ingresados por el usuario y se envían al servicio de autenticar, como se observa en la siguiente figura.

Figura 71. Conexión Android a servicio Autenticar 2

```
SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
request.addProperty( name: "usuario", nro1.getText().toString());
request.addProperty( name: "contrasena", nro2.getText().toString());

SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);
envelope.setOutputSoapObject(request);
//envelope.dotNet= true;

try {
    HttpTransportSE ht = new HttpTransportSE(URL);
    ht.call(SOAP_ACTION, envelope);
    Object resultRequestSOAP = (Object) envelope.getResponse();
    //SoapPrimitive response = (SoapPrimitive) envelope.getResponse();
    resultado=resultRequestSOAP.toString();
    Log.i( tag: "Resultado: ", resultado);
}
catch (Exception e)
{
    Log.i( tag: "Error: ", e.getMessage());
    e.printStackTrace();
    return false;
}

return true;
}
```

Fuentes. Autores.

Después de realizar todas las configuraciones lógicas, se procede a ejecutar el proyecto Android, donde se ingresa los datos de un usuario de la base de datos para probar el debido funcionamiento de la conexión con el servidor y el servicio de Autenticar.

Figura 72. Pantalla servicio autenticar entrada



Fuentes. Autores

Finalmente, al ingresar los datos correctos del usuario, se obtuvo como salida los datos del usuario ingresado, así comprobando que la conexión entre el servidor y los servicios fue exitosa, así demostrando y dejando como guía, que con esta implementación se puede llevar a cabo los demás servicios.

Figura 73. Salida del servicio autenticar en consola de Android

```
I/Resultado:: anyType{persona=anyType{apellidos=Guerrero Alarcón; clave=Rq2eqzgFBjK  
.6GRx8SwLz1FsZIep5K.; codinstitucion=5; codpersona=1; codrol=1; correo=anguerrco@msn.com;  
direccion=calle 2 N 5-76; documento=73167775;  
fechaexpediciondocumento=1974-11-04T00:00:00:00:00; genero=1; nombres=Carlos Andrés;  
telefono=301-7900-437; tipodocumento=CC; usuario=anguerrco@msn.com; }; }  
W/IIInputConnectionWrapper: getExtractedText on inactive InputConnection  
W/IIInputConnectionWrapper: getTextBeforeCursor on inactive InputConnection  
W/IIInputConnectionWrapper: finishComposingText on inactive InputConnection  
getExtractedText on inactive InputConnection  
W/IIInputConnectionWrapper: getTextBeforeCursor on inactive InputConnection  
|
```

Fuentes. Autores.

Ruta de repositorio:

<https://github.com/Edgar1121830/ServicesUsta.git>

7. CONCLUSIONES

Como resultado del catálogo de servicios se definieron las API's que ofrecen la plataforma de inglés para el apoyo en la formación del nivel A1, con el fin de poner en práctica los temas aprendidos en cuanto a lenguaje de programación Java y XML y así presentar los servicios establecidos.

Con el análisis y la introspección del código de la plataforma de inglés, se logró la identificación y definición de los servicios web, teniendo como resultado el mapa conceptual, el cual fue de gran ayuda para la construcción del catálogo y la reutilización del código, que en nuestro caso fue la creación de los DTO.

El modelo 4+1 vistas de Krutchen [1], apoyó al desarrollo del proyecto en el modelamiento de cada uno de los componentes propuestos, así demostrar una estructura lógica y ordenada en la construcción del catálogo de servicios propuesto.

Posteriormente, se llevó a cabo la construcción de los cascarones de los servicios a exponer y se desarrolló uno de estos con el fin de probar su funcionamiento entre el servicio y la aplicación móvil, y así los desarrolladores de la plataforma Web tengan un lineamiento para la construcción de los servicios faltantes.

De igual forma, se realizó la implementación y el despliegue del catálogo, el cual fue exitoso y se ubicó dentro del componente administrador de la plataforma de inglés, donde los desarrolladores podrán tener acceso a todo el catálogo.

8. ANÁLISIS DE RESULTADOS

Para el diseño de los servicios definidos en el catálogo, se utilizó el modelo 4+1 vistas, porque permite diseñar una arquitectura sólida para cualquier aplicación con el fin de obtener un producto de calidad.

Con la arquitectura desarrollada con componentes, se utilizaron tecnologías que facilitan entender la composición de un proyecto web, des modularizado y especializado para la construcción del catálogo de servicios.

Finalmente, se realizó el catalogo de los servicios donde se encuentra la recopilación de la información de este documento, el cual servirá de línea base para los desarrolladores, así facilitarles la comprensión de la arquitectura y funcionamiento del mismo.

9. REFERENCIAS

- [1] «Eclipse Foundation,» [En línea]. Available: <https://www.eclipse.org/org/>. [Último acceso: 30 10 2018].
- [2] «Apache Maven Project,» [En línea]. Available: <https://maven.apache.org/>. [Último acceso: 30 10 2018].
- [3] «NetBeans IDE,» [En línea]. Available: <https://netbeans.org/features/index.html>. [Último acceso: 30 10 2018].
- [4] «JBoss Developer,» [En línea]. Available: <https://developer.jboss.org/welcome>. [Último acceso: 30 10 2018].
- [5] «MySQL,» [En línea]. Available: <https://www.mysql.com/>. [Último acceso: 30 10 2018].
- [6] «Cacoo,» [En línea]. Available: <https://cacoo.com/es/features>. [Último acceso: 30 10 2018].
- [7] «Oracle,» [En línea]. Available: https://docs.oracle.com/cd/E16439_01/doc.1013/e13981/pkgapp001.htm. [Último acceso: 30 10 2018].
- [8] «SoapUI,» [En línea]. Available: <https://www.soapui.org/resources.html>. [Último acceso: 30 10 2018].
- [9] R. Moya, «Jarroba,» [En línea]. Available: <https://jarroba.com/modelo-41-vistas-de-kruchten-para-dummies/>. [Último acceso: 30 10 2018].

10. ANEXOS

A continuación, se podrá observar los anexos, con información adicional requerida para el desarrollo del catálogo de servicios de la USTA.

Anexo 1. Entrevista dirigida al encargado de la plataforma



Entrevista dirigida al encargado de la plataforma

Nombre: Carlos Andrés Guerrero Alarcón
Compañía/Organización: Universidad Santo Tomas Seccional Tunja
Cargo: Decano Facultada Sistemas Fecha: 10/08/2018.

1. ¿Qué entregables brindará a la oficina de idiomas?
Para el Departamento de idiomas será entregado el aplicativo web, en donde los participantes inscritos, podrán navegar a la plataforma creada.
2. ¿Qué importancia le daría usted a esta aplicación?
La importancia en desarrollar y entregar este producto es de alta, debido que es un aplicativo generado por la facultad de sistemas.
3. ¿Cuáles son sus expectativas sobre la confiabilidad de la aplicación?
La aplicación debe ser altamente confiable, debe estar disponible 7*24, de esta forma los estudiantes podrán tener acceso a ella en todo momento que lo requiera, de esta manera estar estudiante todo el tiempo que lo requiera.
4. ¿Cuáles son sus expectativas sobre la capacidad (rendimiento) de la aplicación?
El rendimiento debe y tiene que ser alto, y debe ser concurrente y disponible, esto garantiza un buen trabajo por parte de los estudiantes de la facultad de ingeniería de Sistemas.

REQUERIMIENTOS

1. ¿Qué plataformas se utilizan para estas aplicaciones? Las plataformas en cuanto a desarrollo que serán utilizadas son: como motor de base de datos MySQL y como editor de código php.
2. ¿Usted le dará Soporte a la aplicación o lo hará alguien más? Yo daré soporte a la aplicación.

3. ¿Cuál será el nivel de acceso para el mantenimiento y servicio? El nivel de acceso para el mantenimiento del servicio será apoyado por parte del grupo de docentes y por parte del servicio existirá un encargado por parte del centro de idiomas, el cual será debidamente capacitado en el manejo y funcionamiento de la aplicación.

4. ¿Cómo será distribuida la aplicación? La aplicación será desplegada en el servidor de la universidad, de esta forma podrá ser acceda bajo el dominio de esta misma.

Anexo 2. Plantilla de Requerimientos

A continuación, se podrá observar detalladamente, las plantillas de requerimientos por cada uno de los servicios brindados por la plataforma.

Tabla 8. Caso de uso iniciar sesión

ID: CU001
Nombre: Iniciar Sesión
Autor: Yesica Daniela Vargas Espitia, Andrés Mauricio Pérez Escobar & Andrés Felipe Herrera Rodríguez
Fecha: 15/08/2018
Descripción: En este caso de uso, el usuario podrá ingresar a la plataforma con su respectivo usuario y contraseña (correo electrónico inscrito).
Actores: Administrador y Usuario
Precondiciones: El usuario debe existir en la base de datos
Flujo normal: <ul style="list-style-type: none">- Digitar credenciales de logueo: usuario (cadena de texto con formato de correo electrónico) y contraseña (cadena de texto alfanumérica).- Verificación de credenciales.- Ingresar a la plataforma
Flujo alterno: <ul style="list-style-type: none">- Credenciales inválidas.
Postcondiciones: Listar cursos, ver perfil, ver mis cursos y ver mis certificados.

Fuente. Autores.

Tabla 9. Caso de uso Listar cursos

ID: CU002.1
Nombre: Listar cursos
Autor: Yesica Daniela Vargas Espitia, Andrés Mauricio Pérez Escobar & Andrés Felipe Herrera Rodríguez
Fecha: 15/08/2018
Descripción: Una vez el usuario accede a la plataforma, podrá observar la lista de los cursos del idioma inglés, según el nivel avanzado.
Actores: Administrador y Usuario
Precondiciones: CU001
Flujo normal:

- Ver los cursos
Postcondiciones: Cargar subtemas.

Fuente. Autores.

Tabla 10. Caso de uso cargar subtemas

ID: CU002.1.1
Nombre: Cargar Subtemas
Autor: Yesica Daniela Vargas Espitia, Andrés Mauricio Pérez Escobar & Andrés Felipe Herrera Rodríguez
Fecha: 15/08/2018
Descripción: Cuando el cliente ingrese a la plataforma y seleccione el curso, observará la lista de subtemas, ejemplo: THIS IS MY LIFE, THIS IS MY FAMILY & LISTEN.
Actores: Administrador y Usuario
Precondiciones: CU002.1
Flujo normal: <ul style="list-style-type: none">- Ver cursos- Seleccionar curso- Ver los subtemas
Postcondiciones: Listar actividades.

Fuente. Autores.

Tabla 11. Caso de uso Listar actividades

ID: CU002.1.1.1
Nombre: Listar Actividades
Autor: Yesica Daniela Vargas Espitia, Andrés Mauricio Pérez Escobar & Andrés Felipe Herrera Rodríguez
Fecha: 15/08/2018
Descripción: Después de que el usuario elige el subtema, se carga la lista de actividades a desarrollar, como, por ejemplo: PHASES OR ACTIVITIES.
Actores: Administrador y Usuario
Precondiciones: CU002.1.1
Flujo normal: <ul style="list-style-type: none">- Seleccionar subtema- Ver las actividades
Postcondiciones: Ver contenido temático.

Fuente. Autores.

Tabla 12. Caso de uso ver contenido temático

ID: CU002.1.1.1.1
Nombre: Ver Contenido Temático
Autor: Yesica Daniela Vargas Espitia, Andrés Mauricio Pérez Escobar & Andrés Felipe Herrera Rodríguez
Fecha: 15/08/2018
Descripción: Una vez el usuario elige la actividad, podrá visualizar el contenido, las tareas y las prácticas a desarrollar.
Actores: Administrador y Usuario
Precondiciones: CU002.1.1.1
Flujo normal: <ul style="list-style-type: none">- Seleccionar actividad- Ver el contenido
Postcondiciones: Listar cursos.

Fuente. Autores.

Tabla 13. Caso de uso ver perfil de usuario

ID: CU003.1
Nombre: Ver Perfil de usuario
Autor: Yesica Daniela Vargas Espitia, Andrés Mauricio Pérez Escobar & Andrés Felipe Herrera Rodríguez
Fecha: 15/08/2018
Descripción: Cuando el usuario esté registrado y logueado, podrá acceder a la información de su perfil, para poder actualizar o editar sus datos personales.
Actores: Administrador y Usuario
Precondiciones: CU001
Flujo normal: <ul style="list-style-type: none">- Ver perfil- Actualizar información qué datos van y como son (Opcional)- Foto (array), Documento (cadena de tipo numérico), tipodocumento (cadena de tipo numérico), fechaexpedicióndocumento (cadena de tipo Date), nombre(cadena de texto), Apellidos(cadena de texto), correo(cadena de texto con formato de correo electrónico), género(boolean), telefono(cadena de tipo numérico) y direccion(cadena de texto alfanumérico).
Postcondiciones: Volver a página principal de sesión.

Fuente. Autores.

Tabla 14. Caso de uso ver mis cursos

ID: CU004.1
Nombre: Ver Mis Cursos
Autor: Yesica Daniela Vargas Espitia, Andrés Mauricio Pérez Escobar & Andrés Felipe Herrera Rodríguez
Fecha: 15/08/2018
Descripción: En este caso de uso, el usuario podrá observar qué cursos ha finalizado y cuales están en proceso.
Actores: Administrador y Usuario
Precondiciones: CU001. El usuario debe tener cursos en proceso o finalizados.
Flujo normal: - Ver mis cursos colocar los cursos que están en proceso.
Postcondiciones: Volver a inicio.

Fuente. Autores.

Tabla 15. Caso de uso mostrar certificados

ID: CU005.1
Autor: Yesica Daniela Vargas Espitia, Andrés Mauricio Pérez Escobar & Andrés Felipe Herrera Rodríguez
Fecha: 15/08/2018
Descripción: El cliente podrá observar qué certificados ha obtenido según la terminación y aprobación de los cursos realizados.
Actores: Administrador y Usuario
Precondiciones: CU001. El usuario debió finalizar por lo menos un curso brindado por la aplicación.
Flujo normal: - Ver mis certificados - Mostrar certificado
Postcondiciones: Volver a inicio.

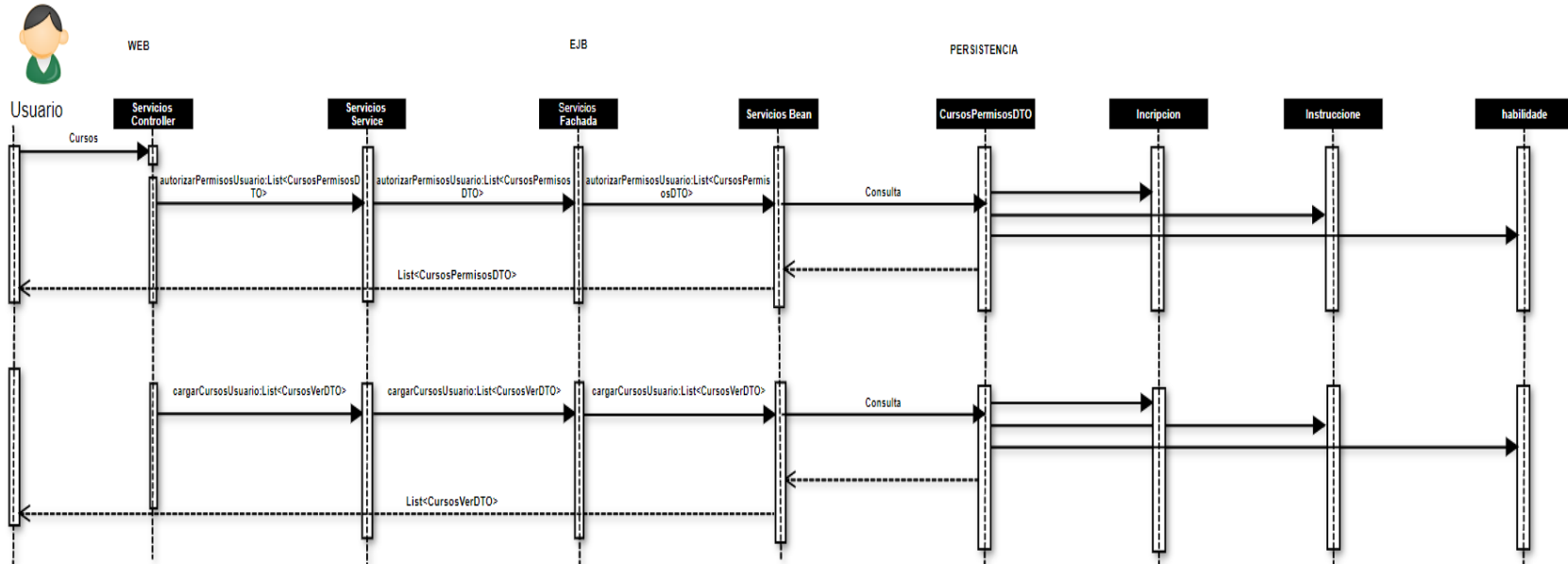
Fuente. Autores.

Anexo 3. Mapa Conceptual



Anexo 4. Diagrama de Secuencia Cursos Permitidos

En este anexo se podrá visualizar el diagrama de secuencias para el servicio de los cursos permitidos.



Anexo 5. Diagrama de Secuencia Actualizar Curso

A continuación, se podrá observar el diagrama de secuencias del servicio actualizar curso.

