****

**UNIVERSIDAD AUTÓNOMA DE TLAXCALA**

UNIDAD ACADÉMICA MULTIDISCIPLINARIA

CAMPUS CALPULALPAN

**DESARROLLO DE UN SISTEMA WEB PARA EL APRENDIZAJE Y EVALUACIÓN DE CONOCIMIENTOS SOBRE EL LENGUAJE JAVA UTILIZANDO TECNOLOGÍAS JSFs Y JAVA COMPILER API**

**“ SEA"**

**TESIS**

QUE PARA OBTENER EL GRADO DE LICENCIATURA:

**INGENIERO EN COMPUTACIÓN**

PRESENTA:

**C. Alfonso Barrera García**

CALPULALPAN, TLAX. Septiembre de 2015

Desarrollo de un Sistema Web para el Aprendizaje y Evaluación de conocimientos sobre el lenguaje Java utilizando tecnologías JSFs y Java Compiler API “SEA”.

Tesis Presentada a la

Unidad Académica Multidisciplinaria Campus Calpulalpan

Universidad Autónoma de Tlaxcala

Que para obtener el

Grado:

Ingeniero en Computación

Presenta

C. Alfonso Barrera García

Septiembre 2015

UNIVERSIDAD AUTÓNOMA DE TLAXCALA

UNIDAD ACADÉMICA MULTIDISCIPLINARIA CAMPUS CALPULALPAN

Tesis

Desarrollo de un Sistema Web para el Aprendizaje y Evaluación de conocimientos sobre el lenguaje Java utilizando tecnologías JSFs y Java Compiler API “SEA”.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

C. Alfonso Barrera García.

Aprobada:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[Presidente]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[Secretario]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[Vocal]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ing. Luís Adán Vázquez Jiménez

Director de Tesis

Resumen

La presente tesis detalla el desarrollo de un sistema Web que incluye un compilador online. El cual es una herramienta de apoyo para los alumnos y profesores, dicho sistema cuenta con dos roles uno como maestro y otro como alumno, el primero deberá crear sus propias clases las cuales contendrán un temario y así mismo ejercicios de programación, los cuales deben de cumplir con una estructura establecida como un nombre, una descripción de lo que se pretende resolver, pistas sobre las sentencias o resultados esperados que pueden servir como apoyo extra al estudiante, tiempo y número de intentos para resolver y por último contar con al menos un caso de prueba que se diseñará acorde a las entradas y salidas esperadas en el método para ese ejercicio. Haciendo que el contenido sea dinámico, actualizado y que se ajuste a sus necesidades; por otro lado, el alumno desarrolla el algoritmo que contendrá el método preparado anteriormente por el profesor y que deberá dar solución al ejercicio, para ser ejecutado en tiempo real, al mismo tiempo se generarán reportes del desempeño de cada alumno, los cuales se podrán visualizar o descargar en formato PDF.

Para lograr lo mencionado anteriormente el sistema se apoya de las clases del API Java Compilar que se encuentra integrado en el lenguaje Java; estas clases se encargan de compilar y ejecutar el código “on the fly”, para esto la declaración de los métodos tiene parámetros de entrada y un tipo de retorno (int, float, String, boolean), el sistema toma como base los casos de prueba diseñados por el profesor y valida que de acuerdo a los parámetros de entrada y salida coincidan al momento de ejecutar el código desarrollado por el alumno. El tiempo y número de intentos invertidos en contestar cada ejercicio son almacenados para generar los reportes y tener una métrica de los avances de cada alumno.

Al utilizar los ejercicios planteados en el sistema como tareas en casa o prácticas de autoestudio da la oportunidad para practicar y consolidar la comprensión de los conceptos vistos en la clase de programación específicamente en el lenguaje Java.

Los usuarios pueden encontrar una interfaz intuitiva de fácil navegación, que les brinda la información necesaria para completar cada uno de los ejercicios que se plantean.

Para el desarrollo de éste proyecto se emplearon las siguientes herramientas: NetBeans IDE, lenguaje Java, Java Compiler API, JSFs, PrimeFaces, Apache Tomcat y el administrador de base de datos MySQL.

Agradecimientos

Le agradezco a Dios por haberme acompañado y guiado a lo largo de mi carrera, por ser mi fortaleza en los momentos de debilidad, por darme la fe y esperanza de poder lograr terminar una meta importante en mi vida, y por último agradezco mucho por brindarme una vida llena de experiencias que dejan siempre un aprendizaje y que nunca se olvidarán.

Le doy gracias a mi padre Facundo Barrera García y mis hermanos Rocio, José (que en paz descanse), Joaquín y Bárbara por apoyarme en todo momento, por los valores que me han inculcado y que siempre hay que tener presentes en mi vida, por tener la confianza en mí de terminar la última etapa de mi educación, sobre todo por ser un gran ejemplo de vida a seguir. Gracias por darme estas oportunidades y por valorar mi esfuerzo y tiempo, sin ustedes y sin su apoyo yo no estaría escribiendo este libro que es para ustedes.

Un agradecimiento singular debo al Ingeniero Luis Adán Vázquez Jiménez que, como director de esta tesis, me ha orientado, apoyado y corregido en esta labor con un interés y una entrega que han sobrepasado, con mucho, todas las expectativas que como alumno deposité en su persona.

A mis amigos por confiar y creer en mí y haber hecho de mi etapa universitaria un trayecto de vivencias que nunca olvidaré.

Contenido

Pág.

[Introducción 16](#_Toc476557164)

[Capítulo 1 Preliminares 18](#_Toc476557165)

[1.1 Introducción. 18](#_Toc476557166)

[1.2 Internet. 18](#_Toc476557167)

[1.2.1 Historia del Internet. 18](#_Toc476557168)

[1.3 Lenguajes de programación 20](#_Toc476557169)

[1.3.1 Instrucciones a la computadora 20](#_Toc476557170)

[1.3.2 Lenguaje máquina 21](#_Toc476557171)

[1.3.3 Lenguajes de bajo nivel 21](#_Toc476557172)

[1.3.4 Lenguajes de alto nivel 22](#_Toc476557173)

[1.3.5 Traductores de lenguaje 23](#_Toc476557174)

[1.3.6 La compilación y sus fases 24](#_Toc476557175)

[1.4 El lenguaje de programación Java 26](#_Toc476557176)

[1.4.1 ¿Qué es Java? 26](#_Toc476557177)

[1.4.2 Java como lenguaje de internet 26](#_Toc476557178)

[1.4.3 Java como lenguaje de propósito general 27](#_Toc476557179)

[1.4.4 Características de Java 27](#_Toc476557180)

[1.4.5 Edición, compilación y ejecución de programas 29](#_Toc476557181)

[1.5 PrimeFaces 31](#_Toc476557182)

[1.5.1 Simplicidad y rendimiento 31](#_Toc476557183)

[1.5.2 Facilidad de uso 31](#_Toc476557184)

[1.6 Base de datos 32](#_Toc476557185)

[1.6.1 MySQL 32](#_Toc476557186)

[1.7 ¿Qué es Integrated Development Environment (IDE)? 32](#_Toc476557187)

[1.7.1 NetBeans 33](#_Toc476557188)

[1.8 Servidores 33](#_Toc476557189)

[1.8.1 Apache Tomcat 33](#_Toc476557190)

[1.9 Javax.Tools 34](#_Toc476557191)

[1.10 “On the Fly” 34](#_Toc476557192)

[1.11 JavaCompiler 34](#_Toc476557193)

[1.12 JavaCompiler.CompilationTask 35](#_Toc476557194)

[1.13 JavaFileManager 36](#_Toc476557195)

[1.14 JavaFileObject 36](#_Toc476557196)

[1.15 DiagnosticListener 36](#_Toc476557197)

[1.16 ToolProvider 36](#_Toc476557198)

[1.17 E-learning 37](#_Toc476557199)

[1.17.1 Importancia del aprendizaje a distancia 37](#_Toc476557200)

[1.17.2 La evaluación 37](#_Toc476557201)

[1.17.3 ¿Qué es la evaluación? 38](#_Toc476557202)

[1.17.4 Tipos de evaluación 38](#_Toc476557203)

[1.17.5 Evaluación a distancia y en línea 40](#_Toc476557204)

[1.17.6 Aportaciones 41](#_Toc476557205)

[Capítulo 2 Marco Teórico 42](#_Toc476557206)

[2.1 Introducción 42](#_Toc476557207)

[2.2 CodingBat code practice 42](#_Toc476557208)

[2.2.1 ¿Qué es? 42](#_Toc476557209)

[2.2.2 Contenido 42](#_Toc476557210)

[2.2.3 Beneficios 43](#_Toc476557211)

[2.3 Coding Ground (Online Terminal & IDE) 43](#_Toc476557212)

[2.3.1 Contenido 44](#_Toc476557213)

[2.3.2 Beneficios 44](#_Toc476557214)

[2.4 Grussell (ActiveSQL) 44](#_Toc476557215)

[2.4.1 Contenido 45](#_Toc476557216)

[2.4.2 Beneficios 45](#_Toc476557217)

[2.5 MIT OpenCourseWare (OCW) 45](#_Toc476557218)

[2.5.1 Contenido 46](#_Toc476557219)

[2.5.2 Beneficios 46](#_Toc476557220)

[2.6 CodeSchool TryjQuery 46](#_Toc476557221)

[2.6.1 ¿Qué es? 46](#_Toc476557222)

[2.6.2 Contenido 47](#_Toc476557223)

[2.6.3 Beneficios 47](#_Toc476557224)

[Capítulo 3 Modelado del Sistema 48](#_Toc476557225)

[3.1 Introducción 48](#_Toc476557226)

[3.2 ¿Qué es el Lenguaje Unificado de Modelado (UML)?. 48](#_Toc476557227)

[3.3 Arquitectura MVC. 49](#_Toc476557228)

[3.4 Modelado del sistema Web SEA. 50](#_Toc476557229)

[3.4.1 Diagramas de casos de uso. 50](#_Toc476557230)

[3.4.1.1 SEA-001-00 Crear cuenta. 51](#_Toc476557231)

[3.4.1.2 SEA-002-00 Recuperar contraseña. 53](#_Toc476557232)

[3.4.1.3 SEA-003-00 Inicio de sesión. 54](#_Toc476557233)

[3.4.1.4 SEA-004-00 Administrar clases. 57](#_Toc476557234)

[3.4.1.5 SEA-005-00 Administrar temarios. 58](#_Toc476557235)

[3.4.1.6 SEA-006-00 Administrar ejercicios. 60](#_Toc476557236)

[3.4.1.8 SEA-007-00 Inscribirse a una clase. 63](#_Toc476557237)

[3.4.1.9 SEA-008-00 Resolver ejercicios 64](#_Toc476557238)

[3.4.1.10 SEA-009-00 Administrar avances 66](#_Toc476557239)

[3.4.2 Diagramas de clases 70](#_Toc476557240)

[3.4.2.1 Diagramas de clases 71](#_Toc476557241)

[3.4.3 Diagramas de secuencia 76](#_Toc476557242)

[3.4.4 Diagrama de componentes 86](#_Toc476557243)

[3.4.5 Diagrama de despliegue 87](#_Toc476557244)

[3.5 Interfaz gráfica del sistema SEA 88](#_Toc476557245)

[3.5.1 SEA- CU-001-00 Crear cuenta 89](#_Toc476557246)

[3.5.2 SEA- CU-002-00 Recuperar clave de acceso. 89](#_Toc476557247)

[3.5.3 SEA- CU-003-00 Administrar sesiones. 90](#_Toc476557248)

[3.5.4 SEA- CU-004-00 Administrar clases. 90](#_Toc476557249)

[3.5.5 SEA-CU-005-00 Administrar temario. 91](#_Toc476557250)

[3.5.6 SEA- CU-006-00 Administrar ejercicios. 91](#_Toc476557251)

[3.5.7 SEA- CU-007-00 Inscribirse a una clase. 93](#_Toc476557252)

[3.5.8 SEA- CU-008-00 Resolver ejercicios 93](#_Toc476557253)

[3.5.9 SEA- CU-009-00 Administrar avances 94](#_Toc476557254)

[3.5.10 SEA- CU-009-00 Servicios disponibles. 96](#_Toc476557255)

[3.5.11 SEA- CU-009-00 Modelo de base de datos SEA 97](#_Toc476557256)

[Capítulo 4 Implementación del Sistema 98](#_Toc476557257)

[4.1 Introducción 98](#_Toc476557258)

[4.2 Configuración del ambiente de desarrollo. 98](#_Toc476557259)

[4.2.1 Java Platform JDK. 98](#_Toc476557260)

[4.2.2 IDE NetBeans. 101](#_Toc476557261)

[4.2.3 Gestor de bases de datos MySQL 102](#_Toc476557262)

[4.2.4 Creación de la base de datos. 106](#_Toc476557263)

[4.2.5 Muestreo de tablas en MySQL en ventana de comandos. 107](#_Toc476557264)

[4.3 Para el desarrollo del sistema SEA. 107](#_Toc476557265)

[4.4 Desarrollo de la aplicación. 108](#_Toc476557266)

[4.4.1 Compilación y ejecución del código. 108](#_Toc476557267)

[Capítulo 5 Resultados 112](#_Toc476557268)

[5.1 Introducción 112](#_Toc476557269)

[5.2 Casos de prueba. 112](#_Toc476557270)

[5.2.1 SEA-CP-001-00 Registrar Alumno/Profesor 112](#_Toc476557271)

[5.2.2 SEA-CP-002-00 Recuperar contraseña 113](#_Toc476557272)

[5.2.3 SEA-CP-003-00 Inicio de sesión. 113](#_Toc476557273)

[5.2.4 SEA-CP-004-00 Administrar clases 114](#_Toc476557274)

[5.2.5 SEA-CP-005-00 Administrar temarios. 115](#_Toc476557275)

[5.2.6 SEA-CP-006-00 Administrar ejercicios. 115](#_Toc476557276)

[5.2.7 SEA-CP-007-00 Inscribirse a una clase. 116](#_Toc476557277)

[5.2.8 SEA-CP-008-00 Resolver ejercicios. 117](#_Toc476557278)

[5.2.9 SEA-CP-009-00 Administrar avances. 118](#_Toc476557279)

[Capítulo 6 Conclusiones y trabajo futuro. 120](#_Toc476557280)

[6.1 Conclusiones. 120](#_Toc476557281)

[6.2 Trabajos futuros. 121](#_Toc476557282)

[APÉNDICE A 122](#_Toc476557283)

[A.1 Script de base de datos para el sistema SEA. 122](#_Toc476557284)

[APÉNDICE B 122](#_Toc476557285)

[B.1 Código fuente del Sistema SISE 122](#_Toc476557286)

[B.1.1 Beans 122](#_Toc476557287)

[B.1.2 DAO 124](#_Toc476557288)

[B.1.3 Managed Bean 126](#_Toc476557289)

[B.1.4 Vistas. 128](#_Toc476557290)

[7 Bibliografía 131](#_Toc476557291)

Lista de tablas

[Tabla 1 SEA-001-00 Crear cuenta 1 Registrar Alumno. 52](#_Toc476557292)

[Tabla 2 SEA-001-00 Crear cuenta 2 Registrar Profesor. 53](#_Toc476557293)

[Tabla 3 SEA-002 Recuperar Contraseña. 54](#_Toc476557294)

[Tabla 4 SEA-003 Iniciar sesión (Alumno). 55](#_Toc476557295)

[Tabla 5 SEA-003 Iniciar sesión (Profesor). 56](#_Toc476557296)

[Tabla 6 SEA-003 Cerrar sesión 57](#_Toc476557297)

[Tabla 7 SEA-004 Crear clase. 58](#_Toc476557298)

[Tabla 8 SEA-005 Temario. 60](#_Toc476557299)

[Tabla 9 SEA-006 Ejercicios 61](#_Toc476557300)

[Tabla 10 SEA-006 Crear casos de prueba. 62](#_Toc476557301)

[Tabla 11 SEA-007 Inscribirse a una clase. 64](#_Toc476557302)

[Tabla 12 SEA-008-00 Temario (Alumno) 65](#_Toc476557303)

[Tabla 13 SEA-008-00 Resolver Ejercicio. 66](#_Toc476557304)

[Tabla 14 SEA-009-00 Visualizar gráficas y reportes (Profesor). 67](#_Toc476557305)

[Tabla 15 SEA-009-00 Visualizar gráficas y reportes (Alumno). 68](#_Toc476557306)

[Tabla 16 SEA-009-00 Reporte PDF Alumno. 69](#_Toc476557307)

[Tabla 17 SEA-009-00 Reporte PDF Alumno 70](#_Toc476557308)

[Tabla 18 Servicios Disponibles 97](#_Toc476557309)

[Tabla 19 SEA-CP-001-00 Registrar Alumno/Profesor 113](#_Toc476557310)

[Tabla 20 SEA-CP-002-00 Recuperar contraseña. 113](#_Toc476557311)

[Tabla 21 SEA-CP-003-00 Iniciar sesión. 114](#_Toc476557312)

[Tabla 22 SEA-CP-004-00 Administrar clases. 115](#_Toc476557313)

[Tabla 23 SEA-CP-005-00 Administrar temarios. 115](#_Toc476557314)

[Tabla 24 SEA-CP-006-00 Administrar Ejercicios. 116](#_Toc476557315)

[Tabla 25 SEA-CP-007-00 Inscribirse a una clase. 117](#_Toc476557316)

[Tabla 26 SEA-CP-008-00 Resolver ejercicios. 118](#_Toc476557317)

[Tabla 27 SEA-CP-009-00 Administrar avances. 119](#_Toc476557318)

[Tabla 28 Script de base datos SEA. 122](#_Toc476557319)

[Tabla 29 Bean del sistema SEA. 124](#_Toc476557320)

[Tabla 30 Dao del sistema SEA. 126](#_Toc476557321)

[Tabla 31 Managed Bean del sistema SEA. 128](#_Toc476557322)

[Tabla 32 Vistas del sistema SEA. 130](#_Toc476557323)

Lista de figuras

[Figura 1 Programa ensamblador 22](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557324)

[Figura 2 Intérprete. 23](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557325)

[Figura 3 Compilador de programas. 23](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557326)

[Figura 4 Fases de la compilación 24](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557327)

[Figura 5 Ejecución de un programa. 25](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557328)

[Figura 6 Proceso de compilación a detalle. 25](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557329)

[Figura 7 Transferencia de código fuente Java en Internet. 27](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557330)

[Figura 8 Ejemplo de objeto, clase Cuadrado y dos instancias. 28](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557331)

[Figura 9 Etapas implicadas en compilación y ejecución de un programa. 31](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557332)

[Figura 10 MySQL 32](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557333)

[Figura 11 Arquitectura MVC (Modelo Vista Controlador) 49](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557334)

[Figura 12 Diagrama de casos de uso del sistema. 50](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557335)

[Figura 13 Diagrama de clases SEA-CU-001-00 Crear Cuenta 71](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557336)

[Figura 14 Diagrama de clases SEA-CU-002-00 Recuperar Cuenta. 71](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557337)

[Figura 15 Diagrama de clases SEA-CU-003-00 Iniciar Sesión. 72](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557338)

[Figura 16 Diagrama de clases SEA-CU-005-00 Administrar clases. 72](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557339)

[Figura 17 Diagrama de clases SEA-CU-005-00 Administrar temario. 73](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557340)

[Figura 18 Diagrama de clases SEA-CU-006-00 Administrar ejercicios. 74](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557341)

[Figura 19 Diagrama de clases SEA-CU-007-00 Inscribirse a una clase. 74](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557342)

[Figura 20 Diagrama de clases SEA-CU-008-00 Resolver Ejercicio. 75](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557343)

[Figura 21 Diagrama de clases uso SEA-CU-009-00 Visualizar Gráficas. 76](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557344)

[Figura 22 Diagrama de secuencias SEA-CU-001-00 Crear Cuenta. 77](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557345)

[Figura 23 Diagrama de secuencias SEA-CU-002-00 Recuperar Cuenta. 77](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557346)

[Figura 24 Diagrama de secuencias SEA-CU-003-00 Iniciar Sesión 78](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557347)

[Figura 25 Diagrama de secuencias SEA-CU-004-00 Administrar clases 79](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557348)

[Figura 26 Diagrama de secuencias SEA-CU-004-00 Administrar clases. 80](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557349)

[Figura 27 Diagrama de secuencias SEA-CU-004-00 Administrar clases. 81](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557350)

[Figura 28 Diagrama de secuencias SEA-CU-005-00 Administrar temario. 82](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557351)

[Figura 29 Diagrama de secuencias SEA-CU-006-00 Administrar ejercicios. 83](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557352)

[Figura 30 Diagrama de secuencias SEA-CU-007-00 Inscribirse a una clase. 84](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557353)

[Figura 31 Diagrama de secuencias SEA-CU-008-00 Resolver Ejercicio. 85](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557354)

[Figura 32 Diagrama de secuencias SEA-CU-009-00 Visualizar Gráficas. 86](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557355)

[Figura 33 Diagrama de componentes SISE. 87](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557356)

[Figura 34 Diagrama de secuencias SEA-CU-001-00 Crear Cuenta. 88](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557357)

[Figura 35 interfaz del inicio del sistema SEA 89](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557358)

[Figura 36 Interfaz de usuario SEA-CU-001-00 Crear Cuenta. 89](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557359)

[Figura 37 Interfaz de usuario SEA-002-00 Recuperar clave de acceso. 90](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557360)

[Figura 38 Interfaz de usuario SEA- CU-003-00 Administrar sesiones. 90](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557361)

[Figura 39 Interfaz de usuario SEA- CU-004-00 Administrar clases. 90](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557362)

[Figura 40 Interfaz de usuario SEA-CU-005-00 Administrar temarios 91](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557363)

[Figura 41 Interfaz de usuario SEA-CU-006-00 Administrar Ejercicio. 92](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557364)

[Figura 42 Casos de prueba SEA-CU-001-00 Administrar Ejercicio 92](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557365)

[Figura 43 Interfaz de usuario SEA-CU-007-00 Inscribirse a una clase. 93](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557366)

[Figura 44 Interfaz de usuario SEA-CU-008-00 Resolver ejercicios. 93](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557367)

[Figura 45 Interfaz de usuario SEA-CU-008-00 Resolver ejercicios (Ejercicios). 94](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557368)

[Figura 46 Interfaz de usuario SEA-CU-008-00 Resolver Ejercicio 94](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557369)

[Figura 47 Interfaz de usuario SEA-CU-009-00 Administrar Avances. 95](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557370)

[Figura 48 Interfaz de usuario SEA-CU-009-00 Administrar Avances (Reportes PDF) 95](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557371)

[Figura 49 Interfaz de usuario SEA-CU-009-00 Administrar avances (Gráficas) 96](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557372)

[Figura 50 Diagrama del modelo E-R de SEA 97](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557373)

[Figura 51 JDK 98](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557374)

[Figura 52 Descarga de JDK 99](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557375)

[Figura 53 Instalación de JDK. 99](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557376)

[Figura 54 Configuración de la instalación. 100](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557377)

[Figura 55 Instalación de máquina virtual de java. 100](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557378)

[Figura 56 Instalación finalizada. 100](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557379)

[Figura 57 Descarga de NetBeans 101](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557380)

[Figura 58 Incluir la instalación de Tomcat. 101](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557381)

[Figura 59 Instalación de NetBeans 102](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557382)

[Figura 60 Instalación finalizada. 102](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557383)

[Figura 61 Descarga de MySQL Installer. 103](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557384)

[Figura 62 Instalación e MySQL. 103](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557385)

[Figura 63 Términos de licencia. 103](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557386)

[Figura 64 Tipos de instalación. 104](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557387)

[Figura 65 Progreso de instalación. 104](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557388)

[Figura 66 Configuración de instalación. 104](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557389)

[Figura 67 Asignación de contraseña para usuario root y crear un usuario. 105](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557390)

[Figura 68 Asignación Permisos y contraseña al usuario “seadb”. 105](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557391)

[Figura 69 Finalización de la instalación. 106](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557392)

[Figura 70 Entrando a la Base de datos con el usuario “seadb”. 106](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557393)

[Figura 71 Tablas de la base de datos SEA. 107](file:///H:\MI%20VERSIÓN\Tesis_AlfonsoBarrera_v0.6.docx#_Toc476557394)

# Introducción

En este documento se plasma el trabajo realizado para la elaboración del diseño e implementación del Sistema para la Evaluación y Aprendizaje (SEA) en el lenguaje de programación Java, que incluye un compilador online, dicho sistema es una herramienta de apoyo para alumnos y profesores en el aprendizaje y desarrollo de técnicas de programación; el proyecto fue construido utilizando el patrón de diseño, Modelo-Vista-Controlador (MVC) y tiene como característica principal la disponibilidad y flexibilidad de acceso, factores que influyen en proyectos de ambiente distribuido.

SEA es un sistema que se apoya del API Java Compiler para integrar un compilador con el cual podemos compilar y ejecutar código “on the fly”da dando oportunidad a los profesores de estructurar ejercicios de programación del lenguaje Java que pueden ayudar a reforzar temas que son de dificultad para sus alumnos, Los ejercicios deben contar con una estructura en la cual se basa el sistema para validar los códigos fuente que han desarrollado los alumnos. Los ejercicios comienzan con un título, un problema que describe lo que se pretende obtener como resultado, el tiempo y número de intentos máximo para resolver y su tipo de retorno, una vez que contamos con estas ideas básicas del ejercicio el siguiente paso es crear los casos de prueba que tendrá que superar el código fuente del alumno. En los casos de prueba se pide un resultado/salida que deberá ser acorde al tipo de retorno que se eligió anteriormente, agregaremos bajo qué condiciones (parámetros de entrada) se va a reproducir ese resultado/salida, lo que el profesor hace con estos casos de prueba es generar el cuerpo o firma de un método. Una vez que ya exista un ejercicio el alumno ingresara al sistema razonara y construirá el algoritmo que resuelve el problema planteado y logre superar los casos de prueba sin agotar el tiempo y numero de intentos.

El aprendizaje se basa en la interacción con los usuarios ya que con las funciones del sistema se permite crear una forma de comunicación bidireccional entre el educador y el estudiante, donde el profesor puede crear ejercicios en base al temario de la asignatura de programación y los estudiantes pueden trabajar con ellos y exista una retroalimentación.

El documento se compone de seis capítulos, el primero se divide en dos secciones importantes la primera da una introducción al lector con los conceptos tecnológicos que se manejan dentro del documento para contextualizarlo y después aborda los temas que sustentan esta investigación como lo es e-learning y la evaluación en línea; el capítulo dos presenta trabajos que sirvieron como ideas base para el desarrollo del sistema SEA, algunos son proyectos de tesis y otros sistemas funcionales que se pueden encontrar en internet; en el siguiente capítulo aborda el modelado del sistema basándose en el Lenguaje Unificado de Modelado (UML) para especificar, visualizar y documentar los esquemas empleados dentro del sistema; dado el análisis en este capítulo se prosigue con el objetivo principal de la investigación que es la implementación, explicando el proceso de desarrollo para la ejecución y despliegue del mismo; al finalizar la implementación en el capítulo cinco se muestran los resultados obtenidos durante las pruebas realizadas para tener una apreciación estadística del funcionamiento general del sistema; finalmente se dedica el capítulo seis para expresar las conclusiones correspondientes.

# Preliminares

## Introducción.

En este capítulo se muestran temas que dan un contexto general de los conceptos que se deben conocer para el desarrollo de un sistema, los cuales son de utilidad para el lector de modo que pueda comprender el enfoque que tendrá esta investigación, sabrá conceptos técnicos del lenguaje java, la internet como una red mundial, la importancia de utilizar herramientas para evaluar conocimientos adquiridos en ciertas áreas como la programación, entre otras cosas que son fundamentales tener presentes y así obtener una concepción más clara de las herramientas utilizadas en el sistema que se presenta.

## Internet.

Internet es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP. Ofrece diversos servicios a los usuarios, como pueden ser almacenamiento o distribución de imágenes, audio y video, salas de chat o videoconferencia, aprendizaje electrónico (E-learning), etc. Todos los servicios que ofrece internet son llevados a cabo por miles de ordenadores que están siempre encendidos y conectados a internet, esperando que los usuarios hagan peticiones. (Molina, 2004).

### Historia del Internet.

Los orígenes de Internet se remontan a la década de 1960, como un proyecto de investigación en redes de comunicación de paquetes, dentro de un ámbito militar. A finales de los años sesenta (1969), en plena guerra fría, el Departamento de Defensa Americano llegó a la conclusión de que su sistema de comunicación era demasiado vulnerable

Como alternativa el citado Departamento de defensa, a través de su Agencia de Proyectos de investigación Avanzados (Advanced Research Project Agency, ARPA) decidió estimular las redes de computadoras mediante becas y ayudas a departamentos de informática de numerosas universidades y algunas empresas privadas. Esta investigación condujo a una red experimental de cuatro nodos, que arranco en Diciembre de 1969, se denominó ARPAnet. La idea de esta red era conseguir que la información llegará a su destino, aunque parte de la red estuviera destruida.

La ARPAnet original se convirtió en Internet. Basada en la idea de que haría múltiples redes independientes con un diseño bastante arbitrario, empezando por ARPAnet como red pionera de conmutación de paquetes, pero que pronto incluiría redes de paquetes de satélite, redes terrestres de radio paquetes y otras redes. Internet tal y como lo conocemos hoy en día plasma una idea técnica subyacente fundamental, que es la red de arquitectura abierta. Cuya principal característica reside en fragmentar la información, dividirla en porciones de una determinada longitud a las que se llama paquetes. Cada paquete lleva asociada una cabecera con datos referentes al destino, origen, códigos de comprobación etc. Así el paquete contiene información suficiente como para que se le vaya encaminando hacia su destino en los distintos nodos que atraviese. El camino a seguir, sin embargo, no está preestablecido de forma que, si una parte de la red cae o es destruida, el flujo de paquetes será automáticamente encaminado por nodos alternativos. Los códigos de comprobación permiten conocer la pérdida o corrupción de paquetes, estableciéndose un mecanismo que permite la recuperación. El sistema de transmisión reúne múltiples ventajas.

* **Fiabilidad,** independiente de la cantidad de las líneas utilizadas y las caídas de la red.
* **Distribución más fácil de los datos** dado que al contener cada paquete la información necesaria para llegar a su destino, tenemos que paquetes con distinto objetivo pueden compartir un mismo canal o camino de comunicación.
* **Posibilidad de técnicas de compresión** que aumentan la capacidad de transmisión y de encriptado que permiten una codificación, de forma que se asegure la confidencialidad de los datos.

Al igual que los equipos o las conexiones también se evolucionó en los servicios que ofrecía ARPAnet, ya que si bien al principio permitía ejecutar programas en modo remoto, en 1972 se introdujo un sistema de correo electrónico, que liberó a los usuarios de la dependencia de los husos horarios (algo de importancia evidente en Estados Unidos, por su gran extensión), y supuso un sorprendente aumento en el tráfico generado, convirtiéndose en la actividad que mayor volumen generaba, en contra de las previsiones iniciales.

Para que las computadoras puedan comunicarse entre sí es necesario que todas ellas envíen y reciban la información de la misma manera. La descripción de los pasos a seguir se denomina “protocolo”. En 1974, se presentó “Transmission Control Protocol / Internet Protocol” (TCP/IP). Este protocolo proporcionaba un sistema independiente de intercambio de datos entre las computadoras y redes locales de distinto origen, eso sí, conservando las ventajas relativas a la técnica de conmutación de paquetes.

A principios de los ochenta el Departamento de Defensa de los Estados Unidos decidió usar el protocolo TCP/IP para la red ARPAnet, desdoblándola en ARPAnet bajo un tráfico extremadamente controlado. Igualmente en Europa se creó la red Minet, como extensión de Milnet..

Dado que una gran cantidad de organismos tenían sus propias redes de área local (RAL) conectadas a los nodos de la red, fue evolucionando hacia una red llamada ARPA Internet formada por miles de equipos. El nombre sufrió algún cambio más como: Federal Research Internet, TCP/IP Internet y finalmente, **INTERNET.**

Durante los últimos años Internet creció hasta incluir el potencial informático de las universidades y centros de investigación, lo que unido a la posterior incorporación de empresas privadas, organismos públicos y asociaciones de todo el mundo supuso un fuerte impulso para **Internet** que dejó de ser un proyecto con protección estatal para convertirse en la mayor red de computadoras del mundo.

## Lenguajes de programación

Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras.

Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un programa informático se le llama programación.

### Instrucciones a la computadora

Los diferentes pasos (acciones) de un algoritmo se expresan en los programas como instrucciones, sentencias o proposiciones(normalmente el término instrucción se suele referir a los lenguajes máquina o lenguajes de bajo nivel, reservando la sentencia o proposición para los lenguajes de alto nivel). Por consiguiente, un programa consta de una sentencia de instrucciones, cada una de las cuales especifica ciertas operaciones que debe ejecutar la computadora.

La elaboración de un programa requerirá conocer el juego o repositorio de instrucciones del lenguaje. Las instrucciones básicas y comunes a casi todos los lenguajes de programación se pueden condensar en cuatro grupos.

* **Instrucciones de entrada/salida.** Instrucciones de transferencia de información y datos entre dispositivos periféricos (teclado, impresora, unidad de disco, etc.) y la memoria central.
* **Instrucciones aritmético-lógicas.** Instrucciones que ejecutan operaciones aritméticas (suma, resta, multiplicación, división, potenciación), lógicas (operaciones and, or, not, etc.).
* **Instrucciones selectivas.** Instrucciones que permiten la selección de tareas alternativas en función de los resultados de diferentes expresiones condicionales.
* **Instrucciones repetitivas.** Instrucciones que permiten la repetición de secuencias de instrucciones un número determinado de veces.

### Lenguaje máquina

Los lenguajes máquina son aquellos que están escritos en lenguajes directamente inteligibles por la computadora, ya que sus instrucciones son cadenas binarias (cadenas o series de caracteres –dígitos– 0 y 1) que especifican una operación y las posiciones (dirección) de memoria implicadas en la operación se denominan instrucciones de máquina o código máquina*.* El código máquina es el conocido código binario.

Las instrucciones en el lenguaje máquina dependen del hardware de la computadora, y por tanto, diferirán de una computadora a otra. En su nivel más fundamental, los únicos programas que se pueden utilizar realmente para operar en una computadora son los programas en lenguaje máquina*.* Tales programas se conocen también *como programas ejecutables* o, simplemente, ejecutables*,* y constan de una secuencia de instrucciones. Las instrucciones están compuestas por números binarios.

Las ventajas de programar en lenguaje máquina son la posibilidad de cargar (transferir un programa a la memoria), sin necesidad de traducción posterior, lo que supone una velocidad de ejecución superior a cualquier otro lenguaje de programación.

### Lenguajes de bajo nivel

Los lenguajes de bajo nivelson más fáciles de utilizar que los lenguajes máquina, pero, al igual que ellos, dependen de la máquina. El lenguaje de bajo nivel por excelencia es el ensamblador (assembly languaje). Las instrucciones en lenguaje ensamblador son instrucciones conocidas como **nemotécnicas** o **nemónicas** (mnemonics). Por ejemplo, nemotécnicos típicos de operaciones aritméticas son: en inglés, ADD, SUB, DIV, etc.; en español, SUM, RES, DIV, etc.

Un programa escrito en lenguaje ensamblador no puede ser ejecutado directamente por la computadora, en esto se diferencia esencialmente del lenguaje máquina, sino que requiere una fase de traducción al lenguaje máquina.

El programa original escrito en lenguaje ensamblador se denomina programa fuente y el programa traducido en lenguaje máquina se conoce como programa objeto*,* ya directamente inteligible por la computadora.

El traductor de programas fuente a objeto es un programa llamado ensamblador (assembler), existente en casi todas las computadoras véase la véase la **Figura 1**. No se debe confundir, aunque en español adoptan el mismo nombre el programa ensamblador está encargado de efectuar la traducción del programa fuente escrito a lenguaje máquina.



Figura 1 Programa ensamblador

### Lenguajes de alto nivel

Los lenguajes de alto nivel son los más utilizados por los programadores. Están diseñados para que las personas escriban y entiendan los programas de un modo mucho más fácil que los lenguajes máquina y ensambladores. Otra razón es que un programa escrito en lenguaje de alto nivel es independiente de la máquina, esto es, las instrucciones del programa de la computadora no dependen del diseño del hardware o de una computadora en particular. En consecuencia, los programas escritos en lenguaje de alto nivel son portables o transportables, lo que significa la posibilidad de poder ser ejecutados con poca o ninguna modificación en diferentes tipos de computadoras, al contrario que los programas en lenguaje máquina o ensamblador, que sólo se pueden ejecutar en un determinado tipo de computadora.

Los lenguajes de alto nivel presentan las siguientes ventajas:

* El tiempo de formación de los programadores es relativamente corto comparado con otros lenguajes.
* La escritura de programas se basa en reglas sintácticas similares a los lenguajes humanos, nombres de las instrucciones tales como READ, WRITE, PRINT, OPEN etc.
* Las modificaciones y puestas a punto de los programas son más fáciles.
* Reducción del coste de los programas.
* Transportabilidad.

Los inconvenientes se concretan en:

* Incremento del tiempo de puesta a punto, al necesitarse diferentes traducciones del programa fuente para conseguir el programa definitivo.
* No se aprovechan los recursos internos de la máquina, que se explotan mucho mejor en lenguajes máquina y ensamblador.
* Aumento de la ocupación de memoria.
* El tiempo de ejecución de los programas es mucho mayor al que sucede con los lenguajes ensambladores.

Los programas fuente tienen que ser traducidos por los programas traductores, llamados en este caso compiladores e intérpretes.

Los lenguajes de programación de alto nivel existentes hoy son muy numerosos, aunque la práctica demuestra que su uso mayoritario se reduce a:

* C, C++, COBOL, Visual BASIC, Java, C#, VB-NET.
* Ada-95, Modula-2, FORTRAN, Pascal, Prolog, LISP, Smalltalk.
* Delphi, Eiffel, Power Builder, SQL.

### Traductores de lenguaje

Los traductores de lenguaje son programas que traducen a su vez los programas fuentes escritos en lenguajes de alto nivel a código máquina. Los traductores se dividen en:

**Intérprete**. Un intérprete es un traductor que toma un programa fuente, lo traduce y a continuación lo ejecuta, véase la **Figura 2.**



Figura 2 Intérprete.

Prácticamente ya no se utilizan, aunque algunas versiones como Qbasic y QuickBASIC pueden todavía tener alguna aplicación en funcionamiento. Sin embargo, está muy extendida la versión interpretada del lenguaje Smalltalk, un lenguaje orientado a objetos puro, Java es un lenguaje interpretado que traduce los códigos de bytes traducidos a su vez por el compilador Java correspondiente, véase la **Figura 3.**

Figura 3 Compilador de programas.

### La compilación y sus fases

La compilación es el proceso de traducción de programas fuente en programas objeto (Aguilar & Zahonero Martínez, 2002). El programa objeto obtenido de la compilación ha sido traducido normalmente código máquina.

Para conseguir el programa máquina real se debe utilizar el programa llamado montador o ligador (linker). El proceso de montaje conduce a un programa en lenguaje máquina directamente ejecutable véase la Figura 4.

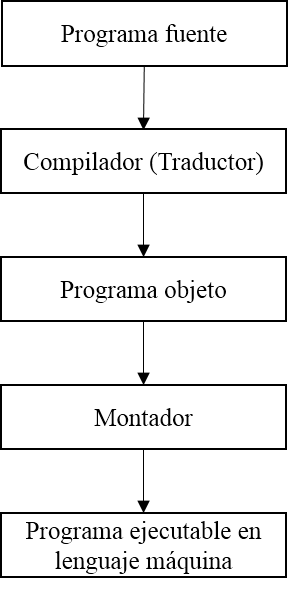


Figura 4 Fases de la compilación

El proceso de ejecución de un programa escrito en un lenguaje de programación y mediante un compilador tiene los siguientes pasos:

1. Escritura del programa fuente con un editor (programa que permite a una computadora actuar de modo similar a una máquina de escribir electrónica) y guardarlo en un dispositivo de almacenamiento.
2. Introducir el programa fuente en memoria.
3. Compilar el programa con el compilador.
4. Verificar y corregir errores de compilación.
5. Obtención de programa objeto.
6. El enlazador (linker) obtiene el programa ejecutable*.*
7. Se ejecuta el programa y, sino existen errores, se tendrá la salida del programa.

El proceso de ejecución seria el mostrado en la Figura 5 y la Figura 6.

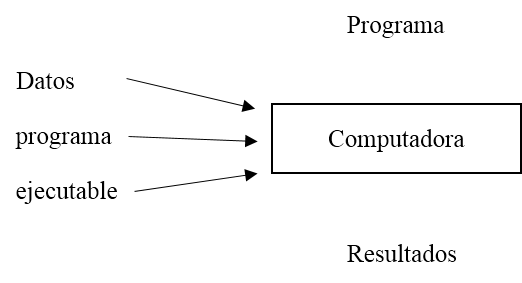


Figura 5 Ejecución de un programa.

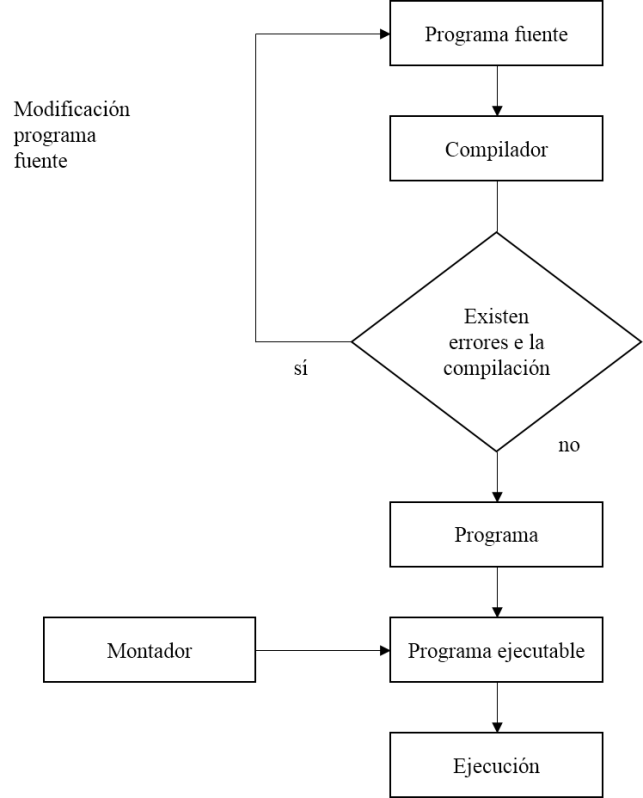


Figura 6 Proceso de compilación a detalle.

## El lenguaje de programación Java

Java es un descendiente de C++ que a su vez es descendiente directo de C. muchas características de Java se han heredado de estos dos lenguajes. De C, Java ha heredado su sintaxis, y de C++, las características fundamentales de programación orientada a objetos.

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban una vez y lo ejecuten en cualquier dispositivo “write once, run anywhere”, esto lo convierte en un lenguaje idóneo para resolver los problemas de seguridad y portabilidad inherentes a los sistemas que trabajan en red. La razón de este aserto reside que en el hecho de que la salida de un compilador Java no es un código ejecutable, sino códigos de bytes (*bytecode*). La traducción de un programa en código de bytes facilita la ejecución del programa en una amplia variedad de entornos y plataformas.

### ¿Qué es Java?

El significado de Java, tal como se le conoce a la actualidad, es el de un lenguaje de programación y un entorno para ejecución de programas escritos en lenguaje Java.

Al contrario que los compiladores tradicionales, que convierten el código fuente en instrucciones a nivel máquina, el compilador Java traduce el código Java fuente en instrucciones que son interpretadas por la Máquina Virtual Java (JVM. Java Virtual Machine). A diferencia de los lenguajes C y C++ en los que está inspirado, Java es un lenguaje interpretado.

Hoy en día Java es uno de los lenguajes de programación más populares para Internet y la World Wide Web en particular, el lenguaje no comenzó como proyecto Internet y por esta circunstancia es idóneo para tareas de programación de propósito general. (Programming Language Popularity, 2013)

### Java como lenguaje de internet

Java es un lenguaje para programar en Internet que trata de resolver problemas claves con el contenido de Internet.

En el mundo de la Web, Java es unatecnología facilitadora que permite a los desarrolladores crear páginas web que se entregarán de modo consistente a todos los usuarios con un navegador habilitado para Java y con independencia de la plataforma hardware y el sistema operativo que se está utilizando.

Dado que el código nativo se interpreta, si existe un intérprete Java para la plataforma específica de hardware o sistema operativo, se pueden escribir programas con el conocimiento de que serán útiles en esa plataforma.

La **Figura 7** muestra como el código nativo Java se transfiere en Internet. En la computadora servidor (host) se almacena el código nativo. Cuando un usuario de un computador local se conecta con el servidor a través de Internet mediante un navegador habilitado para Java, el código nativo se transfiere de la computadora servidor a la computadora local.

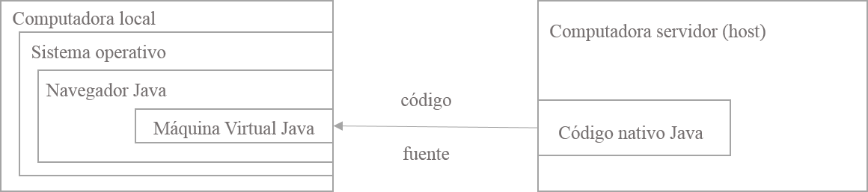


Figura 7 Transferencia de código fuente Java en Internet.

### Java como lenguaje de propósito general

A medida que Java se populariza en desarrollos de Internet gana también como lenguaje de propósito general. Java es totalmente portable a gran variedad de plataformas hardware y sistemas operativos. Java tiene muchos conceptos de sintaxis de C y C++, especialmente de C++ del que es un lenguaje derivado. Añade a C++ propiedades de gestión automática de memoria y soporte a nivel de lenguaje para aplicaciones multihilo.

Por otra parte, Java, en principio a nivel medio, es más fácil de aprender y más fácil de utilizar que C++, ya que las características más complejas de C++ han sido eliminadas de Java: herencia múltiple, punteros (apuntadores) y sentencia “goto” entre otras.

La implementación de la Máquina Virtual de Java puede ser eficaz y eso hace posible que los programas de Java se ejecuten tan rápidamente como los programas C++. Esta característica clave de Java, unida a sus fortalezas como lenguaje de Internet, convierte al lenguaje en idóneo para desarrollo en sistemas cliente/servidor, soporte masivo de los sistemas informáticos de la mayoría de las empresas y organizaciones.

### Características de Java

Java es un lenguaje interpretado*.* Cuando se escriben programas en Java, en un entorno o en un editor de texto, necesita ser compilado en un conjunto de instrucciones optimizadas denominadas programa. Los creadores de Java escribieron un artículo, ya clásico en el que definían al lenguaje como sencillo, orientado a objetos, distribuido, interpretado, robusto, seguro, arquitectura neutra, alto rendimiento, multihilo y dinámico. (Oracle, 2014)

Además de estas características existen otras dos fundamentales, portabilidad y enfocado a la red Internet:

* **Sencillo:** Java fue diseñado para facilitar las tareas del programador profesional y utilizar eficientemente el lenguaje.
* **Orientado a objetos:** La programación orientada a objetos, modela el mundo real, cualquier cosa del mundo puede ser modelada como un objeto. Así, una circunferencia es un objeto, un automóvil es un objeto, una ventana es un objeto, un libro es un objeto e incluso un préstamo o una tarjeta de crédito son objetos. Un programa Java se denomina orientado a objetos debido a que la programación en Java se centra en la creación, manipulación y construcción de objetos, de acuerdo a la **Figura 8.**

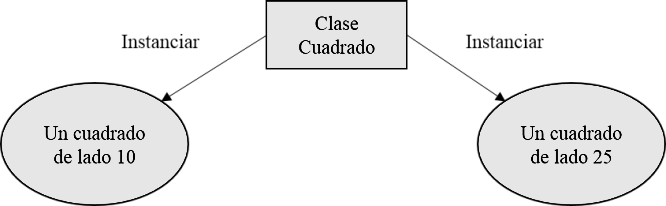


Figura 8 Ejemplo de objeto, clase Cuadrado y dos instancias.

* **Distribuido:** La computación distribuida implica que varias computadoras trabajan juntas en la red. Java ha sido diseñado para facilitar la construcción de aplicaciones distribuidas mediante una colección de clases para uso en aplicaciones en red. La capacidad de red está incorporada a Java, ya que maneja protocolos TCP/IP. De hecho, el acceso a un recurso utilizando un URL es muy similar a un acceso a un archivo.

La escritura de programas en red es similar a enviar y recibir datos a y desde un archivo. La utilización de una URL de Java puede hacer que una aplicación acceda fácilmente a un servidor remoto.

* **Interpretado:** Java es interpretadoy se necesita un intérprete para ejecutar programas Java. Java permite la creación de programas en plataformas cruzadas por compilación en una representación intermedia denominada bytecodeJava: Los programas se compilan en una Máquina Virtual Java generándose un código intermedio denominado bytecode. El bytecode es independiente de la máquina y se puede ejecutar en cualquier máquina que tenga un intérprete Java.
* **Robusto:** Robusto significa fiable. Ningún lenguaje puede asegurar fiabilidad completa, pero java fue ideado con la capacidad para crear programas robustos y a ello se le dio alta prioridad en el diseño. Para ganar en fiabilidad, Java hace restricciones en ciertas áreas clave, para forzar a encontrar errores tempranos en el desarrollo de programas.
* **Seguro:** La seguridad es una característica importante de Java, ya que fue diseñado para ser seguro sin necesidad de comprar software independiente como requieren otros lenguajes de programación. Java permite la construcción de programas libres de virus.
* **Arquitectura neutra:** Una de las principales preocupaciones de los diseñadores de Java es asegurar su longevidad y portabilidad y uno de los grandes problemas a los que se enfrentan los programadores es la falta de garantía de que, si se escribe un programa hoy, mañana funcionará correctamente, incluso en la misma máquina. Las continuas actualizaciones de los sistemas operativos, procesadores y cambios en los recursos fundamentales del sistema se han de combinar para que los programas sean multifunción.
* **Portable:** Java es un lenguaje de alto nivel que permite escribir tanto programas convencionales como aplicaciones para internet (applets). Dado que Internet es una red formada por equipos muy diferentes interconectados por todo el mundo, resulta fundamental para los programas que rueden en ella su independencia de la plataforma en la que van a ser ejecutados.
* **Alto rendimiento:** Los compiladores de Java han ido mejorando sus prestaciones en las sucesivas versiones. Los nuevos compiladores conocidos como JIT (Just In Time) permiten que programas Java independientes de la plataforma se ejecuten con casi el mismo rendimiento en tiempo de ejecución que los lenguajes convencionales compilados.
* **Multihilo:** Java es uno de los primeros lenguajes que se han diseñado explícitamente para tener la posibilidad de múltiples hilos de ejecución; es decir, Java es multihilo (multithreading). Multihilo es la capacidad de un programa de ejecutar varias tareas simultáneamente. Los hilos sincronizados son muy útiles en la creación de aplicaciones distribuidas y en red.
* **Dinámico:** Java es un lenguaje interpretado y, en consecuencia, es un lenguaje muy dinámico. En tiempo de ejecución, el entorno Java puede extenderse mediante enlace en clases que pueden estar localizadas en servidores remotos o en una red. Es una gran ventaja sobre lenguajes tradicionales como C++ que enlaza clases antes del momento de la ejecución. Ésta característica hace posible enlazar código dinámicamente de un modo seguro y expeditivo.

### Edición, compilación y ejecución de programas

Antes de que una computadora pueda procesar un programa en un lenguaje de alto nivel, el programador debe introducir el programa fuente en la computadora y la computadora a su vez debe almacenarlo en un formato ejecutable en memoria. Las etapas clásicas en un lenguaje tradicional son: edición, compilación, enlace, ejecución y depuración de un programa. Las herramientas fundamentales empleadas en el proceso de creación de programas son, por tanto: editor, compilador, depurador y naturalmente el sistema operativo.

* **Edición del programa:** La primera etapa en el proceso de puesta a punto de un programa es la introducción o edición del programa. Un programa se crea con un editor y se pone o guarda en un archivo dentro de un directorio o en una carpeta. En general el nombre del archivo que se pone a un programa debe tener el mismo nombre que la clase dónde está definida la extensión .java. por ejemplo, nombrar al archivo fuente con:

HolaMundo.java

* **Compilación del programa:** Una vez creado el archivo HolaMundo.java con un editor o extraído de un trabajo o de una carpeta se debe escribir por teclado.

Javac HolaMundo.java

Dónde después de unos instantes de espera deberá ver de nuevo el indicador de órdenes. Si la compilación ha tenido éxito no sucederá nada de lo contrario aparecerá un mensaje similar a:

“File name specified is not recognized as an internal or external command, operable program or batch file.”

O bien:

Javac: command not found.

* **Ejecución de un programa:** Para ejecutar un programa Java se necesita un segundo archivo. El compilador crea un archivo denominado HolaMundo.class que contiene la versión en bytecode del programa. Como se comentó anteriormente, el código de bytes de Java es la representación intermedia de un programa que contiene instrucciones que ejecutará el intérprete Java.

Un programa compilado con éxito se puede ejecutar por un intérprete tantas cuantas veces se desee sin necesidad de nuevas compilaciones, en la **Figura 9** se muestra el proceso de compilación y ejecución de un programa.

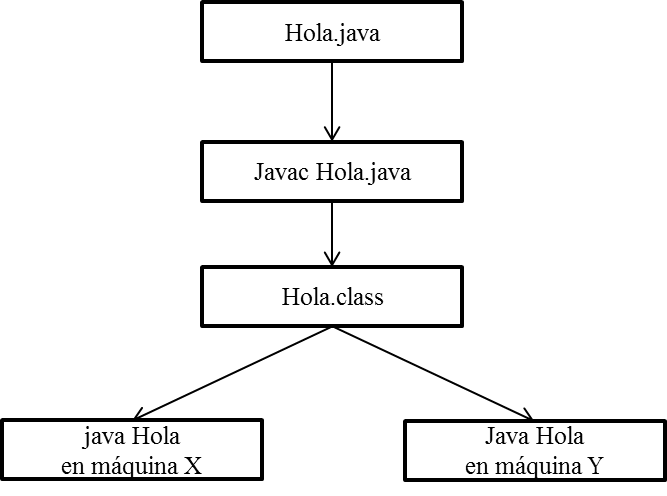


Figura 9 Etapas implicadas en compilación y ejecución de un programa.

## PrimeFaces

Es una librería de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. PrimeFaces está bajo la licencia de Apache License V2. Una de las ventajas de utilizar PrimeFaces, es que permite la integración con otros componentes como por ejemplo RichFaces. En cuanto a experiencia de los usuarios finales los componentes son amigables además que cuentan con un diseño innovador.

### Simplicidad y rendimiento

PrimeFaces es una biblioteca ligera, todas las decisiones tomadas se basan en mantener PrimeFaces lo más ligero posible. Por lo general, la adición de una solución de terceros podría traer una sobrecarga, sin embargo, este no es el caso con PrimeFaces. Es sólo un frasco único, sin dependencias y nada que configurar.

### Facilidad de uso

Los componentes en PrimeFaces se desarrollan con un principio de diseño que establece que "Un buen componente de interfaz de usuario debe ocultar la complejidad, pero mantener la flexibilidad" mientras lo hace.

## Base de datos

Para el diseño de una base de datos se debe establecer un proceso partiendo del mundo real, de manera que sea posible plasmarlo mediante una serie de datos, la imagen que se obtiene del mundo real se denomina modelo conceptual y consiste en una serie de elementos que definen lo que se desea plasmar del mundo real en la base de datos. La definición de este modelo se denomina esquema conceptual.

Una base de datos, en su definición más sencilla, es una colección de archivos relacionados y estructurados en una computadora de manera muy eficiente. Estos archivos pueden almacenar demasiada información que puede ser manipulada y llamada cuando sea necesario. Una base de datos está organizada de forma jerárquica de arriba hacia abajo. Cada tabla está formada por columnas o campos. Los datos están almacenados por filas o registros, el lugar dónde cada fila se intersecta con una columna se conoce como celda. (Kroenke, 2003)

### MySQL

MySQL es un sistema de administración de bases de datos relacionales Open Source y licencia pública, de nivel empresarial y múltiples subprocesos. MySQL fue desarrollado por una firma consultora en Suecia llamada TeX. Ellos necesitaban un sistema de administración de bases de datos que fuera muy rápido y flexible.

Así que crearon MySQL (**Figura 10**) el cual toma algunos conceptos de otro sistema de administración de bases de datos. El producto que ellos crearon es rápido, confiable y demasiado flexible. Es utilizado en muchos lugares y los principales usuarios de MySQL son las universidades, proveedores de servicios de Internet y organizaciones no lucrativas, principalmente por su precio.



Figura 10 MySQL

## ¿Qué es Integrated Development Environment (IDE)?

Un entorno de desarrollo integrado o IDE (en su acrónimo en inglés Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, puede utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, un compilador, un depurador y un constructor de interfaz gráfica (GUI). (Velázquez, 2010)

### NetBeans

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun MicroSystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos (Actualmente Sun Microsystems es administrado por Oracle Corporation).

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

## Servidores

Básicamente un servidor de aplicaciones consiste en un contenedor que abarca la lógica de negocio de un sistema, y provee respuestas a las peticiones de distintos dispositivos que tienen acceso a ella. Son un claro ejemplo del modelo cliente-servidor, cuyo lado cliente ejecuta requerimientos de procesamiento al otro lado, dónde el servidor se encarga de procesar y responder.

### Apache Tomcat

 Funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Oracle Corporation (aunque creado por Sun Microsystems).

Tomcat es un contenedor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

## Javax.Tools

Es un paquete de Java que proporciona las interfaces de las herramientas que se pueden invocar desde un programa, por ejemplo, compiladores. Este paquete es el hogar del lenguaje de programación marco compilador Java. Este marco permite a los clientes la estructura para localizar los compiladores de los programas. El marco también proporciona servicio Provider Interfaces para el acceso estructurado a los diagnósticos (DiagnosticListener), así como un archivo de abstracción para anular el acceso a archivos (JavaFileManager y JavaFileObject).

## “On the Fly”

Java 6 incorpora una API para realizar este tipo de compilaciones dinámicas en tiempo real. Lo que en inglés se conoce con la expresión “On the fly” (al vuelo). Esta compilación dinámica se basa en compilar y ejecutar código fuente Java dentro de una aplicación con código fuente Java.

## JavaCompiler

A partir de JDK 6 viene con varias funcionalidades interesantes, entre ellas la compilación de clases Java desde el mismo java. La instancia que se encarga de tal tarea es JavaCompiler pero su instancia se obtiene desde el método ToolProvider.getSystemJavaCompiler ().

El compilador puede generar diagnósticos durante la compilación (por ejemplo, mensajes de error). Si se proporciona un oyente de diagnóstico, los diagnósticos se suministrarán al oyente. Si no se proporciona ningún oyente, el diagnóstico se dará formato en un formato especificado y se escriben en la salida por defecto, que es System.err a menos que se especifique lo contrario. Incluso si se suministra un oyente de diagnóstico, algunos diagnósticos no pueden caber en un diagnóstico y se escribirá en la salida por defecto.

Una herramienta compilador tiene un administrador de archivos estándar asociado, que es el gestor de archivos que es nativa de la herramienta (o integrado). El administrador de archivos estándar puede ser obtenido llamando getStandardFileManager.

Una herramienta compilador debe funcionar con cualquier gestor de archivos, siempre y cuando se cumplan todos los requisitos adicionales que se detallan en los siguientes métodos. Si no se proporciona ningún gestor de archivos, la herramienta de compilador utilizará un administrador de archivos estándar, como el devuelto por getStandardFileManager.

El compilador se basa en dos servicios: escucha de diagnóstico y administrador de archivos. Aunque la mayoría de las clases e interfaces en este paquete define una API para compiladores (y herramientas en general) las interfaces DiagnosticListener, JavaFileManager FileObject y JavaFileObject no están destinados a ser utilizados en las aplicaciones. En cambio, estas interfaces están destinadas a ejecutarse y se utilizan para proporcionar servicios a medida para un compilador y por lo tanto define un SPI para compiladores.

Hay una serie de clases e interfaces en este paquete, que están diseñados para facilitar la implementación de la SPI para personalizar el comportamiento de un compilador.

## JavaCompiler.CompilationTask

CompilationTask es un objeto que encapsula la operación de compilación actual, esta operación se puede iniciar con la llamada al método call() del mismo objeto. Para obtener un objeto CompilationTask se necesitan pasar ciertos parámetros al método, casi todos los argumentos pueden ser nulos (valores por defecto), pero el argumento final representa la lista de objetos a ser compilados por java así que no puede ser null.

Java CompilerCompilationTask getTask( Write out,

JavaFileManager filemanager,

DiagnosticListener diagnosticListener,

Null,

JavaFileObject fileObjects);

Task.call();

## JavaFileManager

Administrador de archivos que funciona en Java. Gestiona los archivos de entrada y salida, por lo tanto, es responsable de buscar los archivos de entrada y generar los de salida.

Un archivo gestionado por JavaFileManager no necesariamente puede provenir de un archivo físico en disco duro, sino incluso puede venir de memoria o socket remoto, y maneja su contenido en forma de FileObject y JavaFileObject

JavaFileManager no es sólo para JavaCompiler, sino que puede funcionar con cualquier tipo de objetos que se ajusten a la interfaz de la herramienta estándar.

## JavaFileObject

Esta clase proporciona una implementación básica de un objeto archivo que se puede utilizar para su creación, si se desea que el compilador lea el código fuente de algún lugar que no sea un archivo en disco, se utiliza esta clase para definir un objeto archivo almacenado en un String.

## DiagnosticListener

Interfaz para recibir el diagnosticó de las herramientas. Escucha los eventos de diagnóstico de compilación.

Por lo general se refiere a los errores, advertencias o mensajes informativos que puedan aparecer en un programa. Para recibir mensajes de diagnóstico durante el proceso de compilación, podemos adjuntar un Listener al objeto compilador. El Listener es un objeto DiagnosticListener y su método report () será llamado con un método de diagnóstico que contiene mucha más información como el tipo de diagnóstico (error, advertencia, información o cualquier otro), la fuente de este diagnóstico, el número de línea en El código fuente, un mensaje descriptivo etc.

## ToolProvider

Proporciona métodos para la localización de proveedores de herramientas, por ejemplo, los proveedores de compiladores. Esta clase complementa la funcionalidad de ServiceLoader.

Invocación del método run, método heredado de la interfaz Tool (presente desde la versión 6 de Java), es una interfaz utilizada para invocar programas, y en este caso se utiliza para la invocación del compilador. Incluso se puede obtener las versiones soportadas por la herramienta invocando el método getSourceVersions ();

Esta clase proporciona métodos para localizar un objeto Tools. (Recuerde que una herramienta puede ser cualquier cosa como javac, javadoc, rmic, javah, etc.).

## E-learning

El e-learning es la simplificación de Electronic Learning y consiste en la enseñanza y aprendizaje a través de Internet. Este tipo de enseñanza online permite la interacción del usuario con el material mediante la utilización de diversas herramientas informáticas.

Este nuevo concepto educativo es una revolucionaria modalidad de capacitación que posibilito Internet, y que hoy se posiciona como la forma de capacitación predominante. Este sistema ha transformado la educación, abriendo puertas al aprendizaje individual y organizacional.

El e-learning comprende fundamentalmente los siguientes aspectos:

1. El pedagógico, referido a la tecnología educativa como disciplina de las ciencias de la educación, vinculada a los medios tecnológicos, la psicología educativa y la didáctica.
2. El tecnológico, referido a la tecnología de la información y la comunicación, mediante la selección, diseño, personalización, implementación, alojamiento y mantenimiento de las soluciones en dónde se integran tecnologías propietarias y de código abierto (Open Source).

### Importancia del aprendizaje a distancia

Lo más importante en el caso de los alumnos es la flexibilidad en cuanto al manejo de sus propios tiempos, la ausencia del requisito de asistencia periódica a clase, la posibilidad de seguir los estudios desde cualquier parte a dónde el alumno se vea obligado a trasladarse por distintas razones laborales o personales, en definitiva el alto grado de autonomía de la que goza en el sistema, constituye a la educación a distancia en una opción apropiada para los tiempos que corren, ya que permite compatibilizar las exigencias de capacitación con las limitaciones de espacios temporales que impone la vida diaria.

### La evaluación

Proceso dinámico, continuo y sistemático, enfocado hacia los cambios de conducta y/o rendimiento mediante el cual verificamos los logros adquiridos en función de los objetivos propuestos. La evaluación adquiere sentido en la medida que comprueba la eficiencia y posibilita el perfeccionamiento de la acción docente.

No evaluar por evaluar, sino para mejorar los programas, la organización de tareas en línea para mejorar la adquisición de conocimiento mediante una más eficiente selección de las plataformas, contenido e interfaz.

### ¿Qué es la evaluación?

Según el Diccionario de la Real Academia, Evaluación es estimar, apreciar, calcular o señalar el valor de algo.

La evaluación es la determinación sistemática del mérito, el valor y el significado de algo o alguien en función de unos criterios respecto a un conjunto de normas. La evaluación a menudo se usa para caracterizar y evaluar temas de interés en una amplia gama de las empresas humanas, incluyendo las artes, la educación, la justicia, la salud, las funciones y organizaciones sin fines de lucro, los gobiernos y otros servicios humanos. (Pedro, 1983)

Es ligada al desarrollo de exámenes, test y pruebas, su razón de ser ha definido en función de instrumentos o técnicas. El rol fundamental de la evaluación tradicional está centrado en la fase final del proceso de enseñanza – aprendizaje, le interesa determinar la cantidad de conocimiento o contenidos que maneja el alumno durante un tiempo de enseñanza.

### Tipos de evaluación

**Según su finalidad y función**

**Función formativa:** la evaluación se utiliza preferentemente como estrategia de mejora y para ajustar sobre la marcha los procesos educativos, da cara a conseguir las metas o expectativas previstas. Es la más apropiada para la evaluación de procesos, aunque también es formativa en la evaluación de productos educativos, siempre que sus resultados se empleen para la mejor de los mismos. Suele identificarse con la evaluación continua.

**Función sumativa:** suele aplicarse más en la evaluación de productos, es decir, de procesos terminados, con realizaciones precisas y valorables. Con la evaluación no se pretende modificar, ajustar o mejorar el objeto de la evaluación, sino simplemente determinar su valía, en función del empleo que se desea hacer del mismo posteriormente

**Según su extensión**

**Evaluación global:** se pretende abarcar todos los componentes o dimensiones de los alumnos, del centro educativo, del programa, etc. Se considera el objeto de la evaluación de un modo holístico, como una totalidad interactuante, en la que cualquier modificación en uno de sus componentes o dimensiones tiene consecuencias en el resto. Con este tipo de evaluación, la comprensión de la realidad evaluada aumenta, pero no siempre es necesaria o posible.

**Evaluación parcial:** pretende el estudio o valoración de determinados componentes o dimensiones de una institución, de un programa educativo, del rendimiento de unos alumnos, etc.

**Según los agentes evaluadores**

1. **Evaluación interna:** es aquella que es llevada a cabo y promovida por los propios integrantes de una institución, un programa educativo, etc. A su vez, la evaluación interna ofrece diversas alternativas de realización: autoevaluación, hetero-evaluación y coevaluación.
2. **Autoevaluación:** los evaluadores valoran su propio trabajo (un alumno su rendimiento, una escuela o programa su propio funcionamiento, etc.). Los roles de evaluador y evaluado coinciden en las mismas personas.
3. **Heteroevaluación:** evalúan una actividad, objeto o producto, evaluadores distintos a las personas evaluadas (el Consejo Escolar al Claustro de profesores, un profesor a sus alumnos, etc.)
4. **Coevaluación:** es aquella en la que unos sujetos o grupos se evalúan mutuamente (alumnos y profesores mutuamente, unos y otros equipos docentes, el equipo directivo al Consejo Escolar y viceversa). Evaluadores y evaluados intercambian su papel alternativamente.
5. **Evaluación externa:** se da cuando agentes no integrantes de un centro escolar o de un programa evalúan su funcionamiento. Suele ser el caso de la "evaluación de expertos". Estos evaluadores pueden ser inspectores de evaluación, miembros de la Administración, investigadores, equipos de apoyo a la escuela, etc. Estos dos tipos de evaluación son muy necesarios y se complementan mutuamente. En el caso de la evaluación de centro, sobre todo, se está extendiendo la figura del "asesor externo", que permite que el propio centro o programa se evalúe a sí mismo, pero le ofrece su asesoría técnica y cierta objetividad por su no implicación en la vida del centro.

**Según el momento de aplicación**

1. **Evaluación inicial:** se realiza al comienzo del curso académico, de la implantación de un programa educativo, del funcionamiento de una institución escolar, etc. Consiste en la recogida de datos en la situación de partida. Es imprescindible para iniciar cualquier cambio educativo, para decidir los objetivos que se pueden y deben conseguir y también para valorar si al final de un proceso, los resultados son satisfactorios o insatisfactorios.
2. **Evaluación procesual:** consiste en la valoración a través de la recogida continua y sistemática de datos, del funcionamiento de una escuela, de un programa educativo, del proceso de aprendizaje de un alumno, de la eficacia de un profesor, etc. a lo largo del periodo de tiempo fijado para la consecución de unas metas u objetivos. La evaluación procesual es de gran importancia dentro de una concepción formativa de la evaluación, porque permite tomar decisiones de mejora sobre la marcha.

**Evaluación final:** consiste en la recogida y valoración de unos datos al finalizar un periodo de tiempo previsto para la realización de un aprendizaje, un programa, un trabajo, un curso escolar, etc. o para la consecución de unos objetivos.

### Evaluación a distancia y en línea

Permite la adaptación de diversas herramientas para que el usuario logre detectar sus puntos débiles para tratar de corregirlos en aspectos específicos. Cuando las herramientas de evaluación no permiten esta identificación, es preciso la modificación de las estrategias de evaluación diseñadas. El diseñador puede darse cuenta por medio se las sugerencias y observaciones que dejen los usuarios o bien de los mismos resultados.

Para poder evaluar a grandes grupos de personas es necesario contar con herramientas adecuadas y eficientes que garanticen el rigor estadístico para validar una muestra representativa, para ello la evaluación en línea nos ofrecen una amplia gama de ventajas:

**Reducir costos en la aplicación:** Al sustituir materiales físicos por elementos digitales, la aplicación y distribución reducen considerablemente sus costos.

**Acelerar el procesamiento de datos:** Un sistema de encuestas en línea puede ofrecer resultados previos de manera inmediata y en tiempo real, así como mecanismos para ordenar los datos y acelerar los reportes finales.

**Tener un control de usuarios:** Al contar con una base de datos de usuarios, se pueden enviar invitaciones personalizadas a correos electrónicos, se tiene un control preciso de la cantidad de respuestas y permite enviar recordatorios a los usuarios que no han registrado su respuesta.

**Llegar a una gran cantidad de usuarios:** En caso de contar con e-mail, el envío a una gran cantidad de usuarios es sumamente sencillo, inclusive si se encuentran dispersos geográficamente.

**Elegir el nivel de privacidad:** Dependiendo del nivel de privacidad requerido, las respuestas pueden ser identificadas por usuario o simplemente reportar estadísticas generales.

### Aportaciones

Se resaltan tres grandes cambios que la tecnología ha aportado en el contexto de la evaluación, identificamos cada aportación mediante una etiqueta que busca caracterizar el tipo de aportaciones realizado por la tecnología:

La primera aportación es la evaluación automática, en el sentido de que la tecnología contiene bancos de datos que se relacionan entre ellos y se pueden ofrecer a los alumnos respuestas y correcciones inmediatas.

La segunda aportación de las tecnologías en el campo evaluativo se identifica mediante una evaluación del tipo más enciclopédico, en referencia al cúmulo de contenidos que se manejan de una fuente más compleja o de diferentes fuentes. En este caso la ejemplificación más clara se refiere a la elaboración de trabajos monográficos de ensayo sobre una temática específica y para el que se cuenta con Internet como un repositorio de información.

La tercera aportación destacable se refiere a la evaluación colaborativa. En este campo las tecnologías vienen a asistirnos en relación con la visualización de los procesos colaborativos implicados en una evaluación de estas características. Esta aportación tiene diferentes ilustraciones como los debates virtuales, los foros de conservación y los grupos de trabajo.

### 

# Marco Teórico

## Introducción

En los trabajos relacionados que a continuación se presentarán, servirán como base de ideas para el desarrollo de la tesis, ya que algunos de ellos manejan compiladores en línea para ejecutar y compilar código java, otros nos plantean ejercicios para el aprendizaje como se pretende realizar con el sistema que se presenta SEA.

## CodingBat code practice

### ¿Qué es?

CodingBat es un sitio web que nos ofrece problemas de codificación en directo para construir habilidades a la hora de programar en el lenguaje Java y Python, creado por el profesor de informática en Stanford CS. Nick Parlante.

Los problemas de codificación dan retroalimentación inmediata, así que es una oportunidad para practicar y consolidar la comprensión de los conceptos. Los problemas pueden ser utilizados como tarea, o para la práctica de auto-estudio; según Nick Parlante la idea de crear un sitio como este se debe a su experiencia en la enseñanza CS en Stanford combinada con la utilización de unidades de pruebas de los estudiantes, (CodingBat, 2015).

### Contenido

Para comenzar es necesario realizar un registro previo e iniciar sesión en el sitio web CodingBat.com, la página de inicio o principal se divide en dos columnas, a la izquierda encontramos contenido referente al lenguaje de programación Java, del lado derecho se numera un índice de ejercicios en el lenguaje Python; cada una de estas secciones contiene una serie de ejercicios a resolver, entre ellos podemos encontrar temas básicos de Array, String, bucles sentencias de condición como el if, etc. cada una de ellas cubre un tema básico del lenguaje correspondiente; como se mencionó anteriormente al interior de estas secciones se encuentran una serie de ejercicios que abordan el tema seleccionado desde diferentes perspectivas, haciendo que el usuario trabaje en su lógica para dar solución a ellos en diferentes escenarios.

Una vez que se selecciona un ejercicio se puede observar el nombre del ejercicio, a que sección pertenece, una descripción del problema, la caja de texto en la cual se deberá de escribir la lógica que se considere dará solución al problema anteriormente planteado, los casos de prueba que deberá superar nuestra lógica, junto con una gráfica de barras que expresa nuestros avances.

Los ejercicios se encuentran descritos en el idioma inglés, usualmente se narra un problema muy probablemente matemático en el cual se deberán implementar operaciones aritméticas o lógicas que deberán dar con el resultado propuesto por el autor (casos de prueba).

Los errores sintácticos y semánticos que sean generados por el usuario a la hora de codificar el problema se verán reflejados en la misma página, como si se tratará de la salida de un compilador de escritorio (IDE).

El contenido de este sitio web cabe resaltar que es estático, es decir que el autor genero el contenido desde un principio, limitando el enriquecimiento y actualización del contenido. Haciendo que el mantenimiento sea más difícil, además que los ejercicios están diseñados para un público en general y resulta complicado personalizarlo de acuerdo a las necesidades de los usuarios.

### Beneficios

Para sobresalir en Java, o cualquier lenguaje de programación, deberá construir habilidades tanto en lo "grande" y "pequeño". Por "grande" se refiere a las cuestiones amplias, estratégicas de algoritmos, estructuras de datos, etc. lo que pensamos, básicamente, como una licenciatura en Ciencias de la Computación. También necesita habilidad en “pequeño" - 10 o 20 líneas de métodos incorporados de bucles, cadenas, listas, etc. para resolver cada uno de los problemas más grandes.

El número de problemas en cada área es bastante grande, así que se pueden repetir y solidificar habilidades de código en los contenidos que se dificulten más.

La compilación y ejecución del código fuente se hacen en tiempo real, dando respuesta inmediata como si se estuviera en un compilador de escritorio.

## Coding Ground (Online Terminal & IDE)

Coding Ground es un apartado del sitio tutorialspoint.com el cual está dedicado a proporcionar educación de calidad en línea en los dominios de lo académico de Ingeniería Informática, Tecnología de la Información, los lenguajes de programación y de gestión. Esta web ha sido iniciada por ex-alumnos de la UMA, el Sr. Mohtashim, con el lanzamiento de un solo tutorial sobre HTML en el año 2006. En la actualidad, este sitio web es desarrollado y mantenido por Tutoriales Point.

El sitio Web se originó a partir de la idea de que existe una clase de usuarios que responden mejor a los contenidos en línea y prefieren aprender nuevas habilidades a su propio ritmo desde la comodidad de su hogar, los contenidos son de libre acceso y los dueños optan por preservarla a manera de animar a los usuarios a adquirir conocimientos y habilidades en la cantidad que ellos prefieran, (tutorialspoint, 2015)

La misión de este sitio es entregar “Simply Easy Learning” con contenido claro y nítido, y en una amplia gama de temas técnicos y no técnicos sin condiciones previas.

### Contenido

El sitio cuenta con un espacio en el cual el usuario puede crear, editar, compilar y compartir proyectos escritos en el lenguaje Java en la nube. Cuenta con una interfaz muy amigable con el usuario. A diferencia del anterior sitio, este solo tiene la funcionalidad de ser un compilador en línea, en el sitio web no se cuenta con la modalidad de diseño de ejercicios.

Otra de las características que se encuentran es una amplia gama de tutoriales en temas como Big Data, tecnologías de Java, desarrollo Web, bases de datos, tecnologías de Microsoft y XML, etc.

### Beneficios

Algunos de los beneficios que encontramos en este sitio web es el de poder compilar código fuente Java totalmente en línea sin necesidad de instalar plugin o programa en la computadora cliente.

Otro beneficio es el de poder compartir tus proyectos de C, C++, Java, Python, Perl, PHP, etc. En tus redes sociales mediante enlaces simples.

## Grussell (ActiveSQL)

En el sitio db.grussell.org es un sitio que tiene como servicio la ejecución de sentencias (Query) del lenguaje SQL, al igual que los sitios anteriormente mencionados su desarrollo se debió como un experimento para apoyar a los estudiantes a adquirir y reforzar sus conocimientos en este lenguaje.

El Dr. Gordon Russell creador de este sitio es un profesor de la facultad de informática, sus funciones incluyen la docencia a nivel postgrado. Además, está involucrado en la presentación de una serie de cursos CPD para estudiantes y empresas externas, incluyendo cursos a medida en SQL, Bases de datos, Linux, Cisco e Ingeniería de Software. Sus áreas de investigación incluyen entornos integrados de aprendizaje, bases de datos, seguridad de redes y computación en la nube.

### Contenido

ActiveSQL básicamente nos ofrece dos maneras trabajar en su entorno, la primera de manera libre sin estar registrado y la segunda mediante el registro de una cuenta, es visible en los navegadores más comunes; gestiona las respuestas de los usuarios generadas en cada ejercicio para que puedan recordar entre las sesiones, evaluación automática de las consultas SQL con su respectiva información y monitoreo del progreso del usuario.

Actualmente hay cuatro evaluaciones en el sistema. Si se ha completado el 75% de la sección del tutorial, entonces la siguiente sección quedará abierta para el usuario, de lo contrario el siguiente apartado permanecerá cerrado, cabe mencionar que una evaluación cerrada no puede ser alterada después.

Funciones sin registro:

* Se puede trabajar en todos los tutoriales.
* El sistema no recordará sentencias SQL ejecutadas.

Funciones con registro:

* Se puede trabajar en todos los tutoriales.
* El sistema puede recordar las sentencias SQL ejecutadas.
* Puede trabajar en los ejercicios desde cualquier lugar con una computadora que tenga conexión a Internet.
* Se tiene acceso a las evaluaciones.

### Beneficios

Algunos de los beneficios que ofrece el este sitio es el de poder compilar sentencias SQL totalmente en línea sin necesidad de instalar plugin o programa alguno en la computadora cliente; tener un histórico de los avances en cada una de las secciones de evaluación.

## MIT OpenCourseWare (OCW)

Es una plataforma educativa en internet gratuita de la prestigiosa universidad Massachusetts Institute of Technology (MIT),el sitio ofrece más de 2.200 cursos online, que puedes buscar por palabra clave, tema, idioma o incluso por si tienen subtítulos o no. El programafue financiado inicialmente por la fundación William y Flora Hewlettla, Andrew W. Mellon y el MIT. En un principio se servía del sistema de gestión de contenidos basados en Microsoft's Content Management Server, después fue sustituido a mediados del 2010 por un Plone sistema basado en gestión de contenidos.

### Contenido

En el sitio se puede encontrar información más teórica sobre conceptos básicos y avanzados de programación y sus distintos lenguajes.Los educadores pueden mejorar los cursos y programas de estudio, para que sus escuelas sean más eficientes; al ingresar a los **cursos**, a los estudiantes les ayuda a tener un mejor desempeño en sus materias de programación. Algunos cursos también incluyen demostracionesinteractivas Web en Java, Libros de texto completamente escritos por profesores del MIT y video conferencias.

### Beneficios

Como beneficios que encontramos en el sitio Web es que no solo existen ejercicios prácticos de lenguaje de programación, si no que contamos con información teórica, respecto a los temas que se plantean en el curso. Sin dejar de mencionar que todo el contenido que se encuentra publicado cuenta con el respaldo del MIT.

## CodeSchool TryjQuery

La mayoría de la gente no aprende a programar y diseñar para la web mediante la lectura de un libro. El verdadero aprendizaje tiene lugar cuando se empieza a experimentar con el código en el navegador o compilador y el uso de los conceptos básicos. CodeSchool abre las puertas a una nueva forma de aprender mediante la combinación de video, codificación en el navegador, para que el aprendizaje de una nueva tecnología sea diversión.

### ¿Qué es?

CodeSchool idea original de Envy Labs, es una plataforma de aprendizaje en línea que enseña una variedad de programación y conocimientos de diseño web. Los cursos van desde principiantes hasta niveles avanzados, ganando premios e insignias a medida que se aprende. Los cursos que ofrece utilizan screencast y ejercicios interactivos que lo guían hacia un mejor código.

Enseña a través de contenidos de entretenimiento. Cada curso está construido en un entorno creativo para que se sienta como si estuviera jugando y no sentado en un aula. Se combina la mecánica del juego con la instrucción de video y retos en el navegador.

### Contenido

Cada curso tiene por lo menos cinco niveles, cada nivel comienza con un screencast de diez a quince minutos, seguido por una serie de desafíos de código que debe resolver para llegar al siguiente nivel. Una vez que se emplee el curso, se podrá:

* Desbloquear el video final. Esto ayudará en el aprendizaje continuo de la asignatura.
* Recibe una insignia. Que muestra a sus amigos que se ha completado el curso.

Con más de 40 cursos sobre JavaScript, HTML/CSS, Ruby, Git y iOS, los ejercicios están meticulosamente producidos por instructores capacitados, de alta calidad inspirados en los miembros de la red.

Los creadores del sitio hacen mención de contar con la última tecnología así que recomiendan que, para ingresar a los cursos, solo se necesita un navegador web actualizado.

### Beneficios

Las habilidades de codificación nunca han estado en una mayor demanda, pero los cursos de informática tradicionales son a menudo costosos, lentos y carecen de la flexibilidad que la gente demanda, temas que son atacados con estas tecnologías.

Al terminar cada curso el usuario tiene la posibilidad de obtener un tipo boleta o reporte que muestra todo su progreso obtenido durante la duración del curso; también se obtienen insignias que, aunque no tienen un valor económico o de otro tipo, sirve para aparecer en el ranking de la página.

El enriquecimiento y mantenimiento del sitio no solo tarea de los administradores, los usuarios que lo utilizan tienen la opción de reportar errores de validación o contenido que encuentren, con la finalidad de ir mejorando el contenido.

# Modelado del Sistema

## Introducción

Para el desarrollo del sistema será necesario realizar procesos de análisis mediante el Lenguaje de Modelado Unificado (UML) que permitan un mejor entendimiento en el funcionamiento del sistema, es por eso que en este capítulo se muestran las fases de análisis y diseño del sistema mediante diagramas y descripciones de los mismos que nos permiten comprender el funcionamiento del sistema en diferentes vistas.

Para ello lo primero que se mostrará serán las generalidades del Lenguaje de Modelado Unificado (UML), posteriormente se propondrá la arquitectura que se implementará en el sistema del cual se basa el diseño y finalmente se mostrarán los diagramas correspondientes.

## ¿Qué es el Lenguaje Unificado de Modelado (UML)?.

El Lenguaje Unificado de Modelado (Unified Modeling Lenguage, UML) es un lenguaje estándar para escribir planos de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software (wikipedia, 2015).

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y reglas para permitir comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo (workflow) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware.

Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones. En concreto, un diagrama ofrece una vista del sistema a modelar. Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas. UML incluye los siguientes diagramas:

* Diagramas de casos de uso.
* Diagramas de clases.
* Diagramas de objetos.
* Diagramas de secuencia.
* Diagramas de colaboración
* Diagramas de estados.
* Diagramas de actividades.
* Diagramas de componentes.
* Diagramas de despliegue.

## Arquitectura MVC.

El modelo vista controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el modulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos: (wikipedia, 2015)

**Modelo.**

La capa modelo define la lógica del negocio (la base de datos pertenece a esta capa).

**Vista.**

La vista es lo que usan los usuarios para interactuar con la aplicación (los gestores de plantillas pertenecen a esta etapa).

**Controlador.**

El controlador es un bloque de código que realiza llamadas al modelo para obtener datos pasándolo a la vista que serán mostrados al usuario.

Para tener una mejor apreciación sobre la arquitectura MVC a continuación se muestra la Figura 11

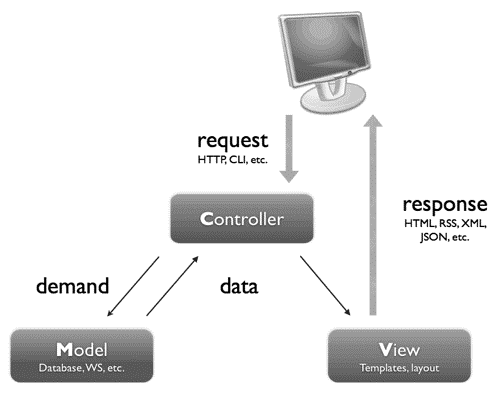


Figura 11 Arquitectura MVC (Modelo Vista Controlador)

## Modelado del sistema Web SEA.

A continuación, en esta sección se verá como modelar la funcionalidad del sistema mediante la representación de diagramas tales como los casos de uso, de secuencias entre otros. En UML los casos de uso son los principales medios para capturar la funcionalidad del sistema desde la perspectiva del usuario.

### Diagramas de casos de uso.

Un caso de uso representa una unidad discreta de interacción entre el usuario y el sistema. Cada caso de uso tiene una descripción, puede incluir también la funcionalidad de otros casos de uso según sea el caso. También el uso de escenarios es importante ya que describen de manera detallada la funcionalidad de un caso de uso.

A continuación en la Figura 12 se muestra el diagrama de casos de uso del sistema.

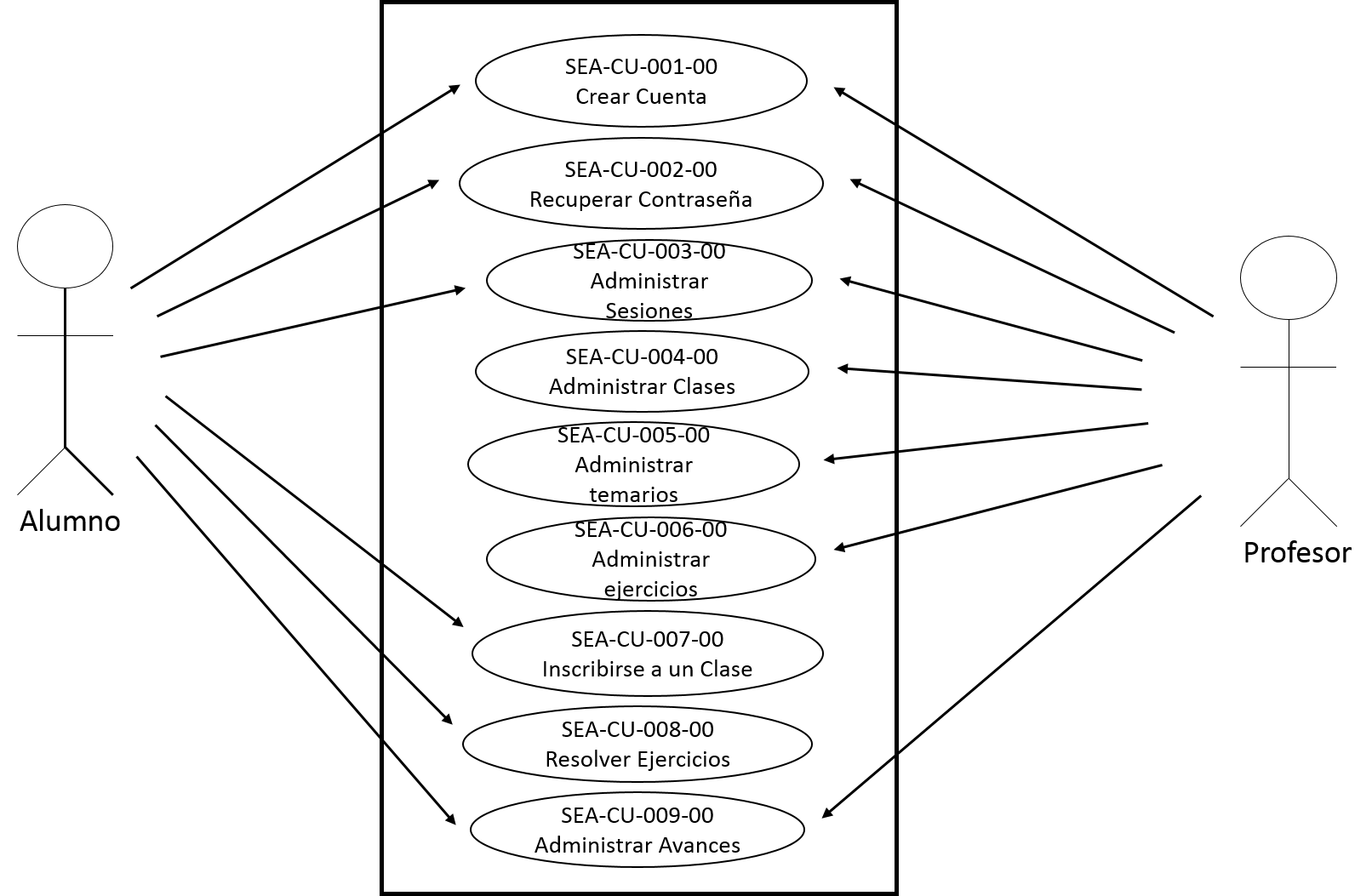


Figura 12 Diagrama de casos de uso del sistema.

### 3.4.1.1 SEA-001-00 Crear cuenta.

Comprende la creación de una cuenta de un actor que funge como “Alumno” o “Profesor” hacia el sistema quien tendrá la finalidad de verificar las credenciales introducidas y ejercer la acción correspondiente.

En la Tabla 1 se muestra el escenario que permite al “Alumno” registrarse al sistema para contar con credenciales que le permitan el acceso al mismo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-001-00 crear cuenta alumno. | |
| **Actores** | | Alumno y Sistema. | |
| **Descripción** | | Permite registrar a los alumnos al sistema. | |
| **Pre condiciones** | | Una cuenta de correo electrónico. | |
| **Post condiciones** | | N/A. | |
| **Escenario 01 Crear Cuenta.** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | Muestra la página principal “Inicio”. | |  |
| 2 |  | | Da clic en la opción “regístrate ahora”, ubicada en la página principal. |
| 3 | Carga el formulario correspondiente y solicita los datos de registro: nombre, apellido paterno, apellido materno, correo, contraseña y una lista de los dos posibles roles como alumno o profesor. | |  |
| 4 |  | | Ingresa la información solicitada para su registro en el sistema: nombre, apellido paterno, apellido materno, correo, contraseña y una lista de los dos posibles roles como alumno o profesor. |
| 5 |  | | Da clic en el botón “Aceptar” para guardar el registro en la Base de Datos de SEA. |
| 6 | En caso de faltar algún campo el sistema muestra el mensaje “**MSJ-01**”. | |  |
| 7 | Almacena los datos.  Enviará un correo electrónico de bienvenida con sus datos de acceso y regresa al menú principal para que acceda con su correo y contraseña. | |  |
|  | **Reglas de negocio** | | |
|  | RN-00101-01 La contraseña debe estar codificada en encriptación MD5. | | |
|  | **Mensajes del sistema** | | |
|  | MSJ-01 El registro no se guardó. | | |

Tabla 1 SEA-001-00 Crear cuenta 1 Registrar Alumno.

En la Tabla 2 se muestra el escenario que permite al “Profesor” registrarse al sistema para contar con credenciales que le permitan el acceso al mismo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-001-00 crear cuenta profesor. | |
| **Actores** | | Profesor y Sistema. | |
| **Descripción** | | Permite registrar al profesor en el sistema. | |
| **Pre condiciones** | | Una cuenta de correo electrónico. | |
| **Post condiciones** | | N/A. | |
| **Escenario 01 Crear Cuenta.** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | Muestra la página principal “Inicio”. | |  |
| 2 |  | | Da clic en la opción “regístrate ahora”, ubicada en la página principal. |
| 3 | Carga el formulario correspondiente y solicita los datos de registro: nombre, apellido paterno, apellido materno, correo, contraseña y una lista de los dos posibles roles como alumno o profesor. | |  |
| 4 |  | | Ingresa la información solicitada para su registro en el sistema. |
| 5 |  | | Da clic en el botón “Aceptar” y para guardar el registro en la Base de Datos de SEA. |
| 6 | En caso de faltar algún campo el sistema muestra el mensaje “**MSJ-01**”. | |  |
| 7 | Almacena los datos.  Enviará un correo electrónico de bienvenida con sus datos de acceso y regresa al menú principal para que acceda con su correo y contraseña. | |  |
|  | **Reglas de negocio** | | |
|  | RN-00101-01 La contraseña debe estar codificada en encriptación MD5. | | |
|  | **Mensajes del sistema** | | |
|  | MSJ-01 El registro no se guardó. | | |

Tabla 2 SEA-001-00 Crear cuenta 2 Registrar Profesor.

### 3.4.1.2 SEA-002-00 Recuperar contraseña.

En la Tabla 3 se muestra el escenario que permite al alumno o profesor obtener de nuevo su clave de acceso olvidado, y poder acceder al sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-002 Recuperar Contraseña. | |
| **Actores** | | Profesor o Alumno y Sistema. | |
| **Descripción** | | Permite recuperar clave de acceso al sistema. | |
| **Pre condiciones** | | Haber creado una cuenta anteriormente en el sistema. | |
| **Post condiciones** | | El profesor o alumno podrá recuperar su clave con el correo que proporciono en el registro. | |
| **Escenario 02 Recuperar Clave de Acceso.** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | Muestra la página principal “Inicio”. | |  |
| 2 |  | | Da clic en la opción ¿Olvidaste tu contraseña?, ubicada en la página principal. |
| 3 | Muestra la ventana Recupera tu cuenta con una caja de texto escribe tu correo, los botones buscar y cancelar. | |  |
| 4 |  | | Ingresa correo y da clic en el botón recuperar. |
| 5 | Verifica si el correo existe y envía la contraseña al correo en caso contrario muestra el mensaje “**MSJ-02**” y regresa a la página principal. | |  |
|  | **Reglas de negocio** | | |
|  | N/A. | | |
|  | **Mensajes del sistema** | | |
|  | MSJ-02 Correo no válido. | | |

Tabla 3 SEA-002 Recuperar Contraseña.

### 3.4.1.3 SEA-003-00 Inicio de sesión.

Comprende el inicio de sesión de un actor como Alumno o Profesor hacia el sistema quien tendrá la finalidad de verificar las credenciales introducidas y ejercer la acción correspondiente.

En la Tabla 4 se muestra el escenario correspondiente que permite al alumno acceder al sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-003 Iniciar Sesión. | |
| **Actores** | | Alumno y Sistema. | |
| **Descripción** | | Permite el acceso del profesor o alumnos al sistema. | |
| **Pre condiciones** | | Haber creado una cuenta anteriormente en el sistema. | |
| **Post condiciones** | | El profesor o alumno podrá acceder al sistema con su correo y contraseñas que proporciono en el registro. | |
| **Escenario 03 Inicio de sesión.** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | Muestra la página principal “Inicio”. | |  |
| 2 |  | | Ingresa correo y contraseña después clic en el botón iniciar sesión. |
| 3 | Verifica que los datos existan en la base de datos de SEA, si existen muestra la página crear clase, caso contrario muestra el mensaje “**MSJ-03**”. | |  |
| 4 | Si los datos son aceptados el sistema muestra una nueva página mi clase (Alumno). | |  |
|  | **Reglas de negocio** | | |
|  | N/A. | | |
|  | **Mensajes del sistema** | | |
|  | MSJ-03 Datos no válidos. | | |

Tabla 4 SEA-003 Iniciar sesión (Alumno).

En la Tabla 5 se muestra el escenario correspondiente que permite al profesor acceder al sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-003 Iniciar Sesión. | |
| **Actores** | | Profesor y Sistema. | |
| **Descripción** | | Permite el acceso del profesor al sistema. | |
| **Pre condiciones** | | Haber creado una cuenta anteriormente en el sistema. | |
| **Post condiciones** | | El profesor podrá acceder al sistema con su correo y contraseñas que proporciono en el registro. | |
| **Escenario 03 Inicio de sesión.** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | Muestra la página principal “Inicio”. | |  |
| 2 |  | | Ingresa correo y contraseña después clic en el botón iniciar sesión. |
| 3 | Verifica que los datos existan en la base de datos de SEA, si existen muestra la página crear clase, caso contrario muestra el mensaje “**MSJ-03**”. | |  |
| 4 | Si los datos son aceptados el sistema muestra una nueva página mi clase (Profesor). | |  |
|  | **Reglas de negocio** | | |
|  | N/A. | | |
|  | **Mensajes del sistema** | | |
|  | MSJ-03 Datos no válidos. | | |

Tabla 5 SEA-003 Iniciar sesión (Profesor).

En la Tabla 6 se muestra el escenario correspondiente que permite al profesor y Alumno cerrar la sesión de su cuenta.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-003 Cerrar Sesión. | |
| **Actores** | | Profesor o Alumno y Sistema. | |
| **Descripción** | | Permite al Profesor o Alumno Terminar la sesión. | |
| **Pre condiciones** | | Haber iniciado sesión en el sistema. | |
| **Post condiciones** | | N/A. | |
| **Escenario 03 Cerrar sesión.** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | Aparece en un menú una vez que se ha iniciado sesión al sistema y permanece visible el botón en todas las ventanas del sistema. | |  |
| 2 |  | | Clic en el botón cerrar sesión. |
| 3 | Termina la sesión de las variables y direcciona a la página principal. | |  |
|  | **Reglas de negocio** | | |
|  | N/A. | | |
|  | **Mensajes del sistema** | | |
|  | N/A. | | |

Tabla 6 SEA-003 Cerrar sesión

### 3.4.1.4 SEA-004-00 Administrar clases.

En la Tabla 7 se muestra el escenario que permite al profesor crear, editar o eliminar una clase.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-004 Crear clase. | |
| **Actores** | | Profesor y Sistema. | |
| **Descripción** | | Permite crear una nueva clase, editar o eliminar una existente. | |
| **Pre condiciones** | | Haber creado una cuenta anteriormente en el sistema. | |
| **Post condiciones** | | N/A. | |
| **Escenario 04 crear clases.** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | Muestra la página “Mi clase”. | |  |
| 2 |  | | Ingresar el nombre de la clase y clic en agregar. |
| 3 | Lista una tabla con la clase creada y las opciones de ver clase, editar o eliminar. | |  |
| 4 |  | | Clic en ver clase. |
| 5 | Direcciona a la ventana “Temario”. | |  |
| 6 |  | | Clic en editar. |
| 7 | Llena el campo de nombre de la clase con los datos de la clase seleccionada, una vez que se edita el campo nombre se actualizará en la BD. | |  |
| 8 |  | | Clic en eliminar. |
| 9 | Muestra un cuadro de diálogo para confirmar la acción, en caso de aceptar se elimina y muestra el mensaje “**MSJ-04**”. | |  |
|  | **Reglas de negocio** | | |
|  | N/A. | | |
|  | **Mensajes del sistema** | | |
|  | MSJ-04 Clase eliminada. | | |

Tabla 7 SEA-004 Crear clase.

### 3.4.1.5 SEA-005-00 Administrar temarios.

En la Tabla 8 se muestra el escenario que permite al profesor crear un temario, temas y subtemas.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-005 Temario. | |
| **Actores** | | Profesor y Sistema. | |
| **Descripción** | | Permite crear temas y subtemas en el temario. | |
| **Pre condiciones** | | Haber creado una clase anteriormente en el sistema. | |
| **Post condiciones** | | N/A. | |
| **Escenario 05 Crear el temario.** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 |  | | Clic en el botón “Ver” de la tabla mis clases en la ventana “Mi clase”. |
| 2 | Direcciona a la página “Temario”. | |  |
| 2 | Muestra la página nuevo tema con una caja de texto y un botón de agregar tema. | |  |
| 3 |  | | Ingresa el nombre del tema y clic en el botón agregar. |
| 4 | Se actualiza la tabla y muestra los temas agregados en formato treeTable. | |  |
| 5 |  | | Clic sobre el tema. |
| 6 | Se despliega un cuadro de diálogo Tema dónde se muestra los datos del tema seleccionado y los botones de agregar subtema, editar tema y eliminar. | |  |
|  |  | | Clic en el botón agregar subtema del cuadro de diálogo Tema. |
|  | Se despliega el cuadro de diálogo subtema dónde se muestra un InputText para el nombre del subtema y el botón de agregar, al finalizar esta acción cerrará los cuadros de diálogo y refrescará la tabla. | |  |
|  |  | | Clic en el botón eliminar del diálogo Tema. |
|  | Muestra un cuadro de diálogo para confirmar la acción, en caso de aceptar se elimina y muestra el mensaje “**MSJ-05**”. | |  |
|  |  | | Clic para desplegar el tema. |
|  | Lista los subtemas de tema correspondiente. | |  |
|  |  | | Clic en el subtema. |
|  | Despliega el cuadro de diálogo subtema dónde se muestra los datos del subtema seleccionado y los botones de eliminar, editar subtema y ejercicios habilitados. | |  |
|  |  | | Clic en el botón eliminar. |
|  | Muestra un cuadro de diálogo para confirmar la acción, en caso de aceptar se elimina y muestra el mensaje “**MSJ-06**”. | |  |
|  |  | | Clic en el botón ejercicios. |
|  | Direcciona a la pantalla Ejercicios. | |  |
|  | **Reglas de negocio** | | |
|  | N/A. | | |
|  | **Mensajes del sistema** | | |
|  | MSJ-05 Tema Eliminado. | | |
|  | MSJ-06 Subtema Eliminado. | | |

Tabla 8 SEA-005 Temario.

### 3.4.1.6 SEA-006-00 Administrar ejercicios.

En la Tabla 9 se muestra el escenario que permite al profesor crear un ejercicio.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-006 Ejercicios. | |
| **Actores** | | Profesor y Sistema. | |
| **Descripción** | | Permite crear ejercicios en cada subtema. | |
| **Pre condiciones** | | Haber creado una clase, temas y subtemas anteriormente en el sistema. | |
| **Post condiciones** | | N/A. | |
| **Escenario 006 Crear un ejercicio.** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | Muestra la página “Ejercicios”. Listando los siguientes campos:  Nombre del problema, Descripción del problema, pista/sugerencia, tiempo máximo, Número de intentos, Código Fuente, una lista del tipo de retorne del método (int, String, float) y el botón agregar Ejercicio. | |  |
| 2 |  | | Llena los campos y clic en agregar ejercicio. |
| 3 | Lista en la tabla lo siguiente:  Nombre del ejercicio y los botones de agregar caso de prueba, editar ejercicio y eliminar. | |  |
| 4 |  | | Clic en agregar eliminar. |
| 5 | Muestra un cuadro de diálogo para confirmar la acción, en caso de aceptar se elimina y muestra el mensaje “**MSJ-07**”. | |  |
| 6 |  | | Clic en agregar casos de prueba. |
| 7 | Dirección a la página casos de prueba. | |  |
|  | **Reglas de negocio** | | |
|  | N/A. | | |
|  | **Mensajes del sistema** | | |
|  | MSJ-07 Ejercicio Eliminado. | | |

Tabla 9 SEA-006 Ejercicios

En la Tabla 10 se muestra el escenario que permite al profesor crear casos de prueba a cada ejercicio.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-006 Crear casos de prueba. | |
| **Actores** | | Profesor y Sistema. | |
| **Descripción** | | Permite agregar casos de prueba a un ejercicio. | |
| **Pre condiciones** | | Que el profesor haya creado una clase. | |
| **Post condiciones** | | Crear un ejercicio. | |
| **Escenario 07 Mis clases** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | Muestra la página “Casos de Prueba” con los siguientes campos: una caja de texto para la salida del ejercicio, el botón de agregar y dos tablas una para las salidas, la segunda tabla listará los parámetros de la salida seleccionada. | |  |
| 2 |  | | Escribe la salida y clic en el botón agregar. |
| 3 | Lista en la tabla la salida que ha agregado con las opciones de agregar parámetros, eliminar, editar y ver parámetros. | |  |
| 4 |  | | Clic en agregar parámetros. |
| 5 | Se visualiza en pantalla un cuadro de diálogo con una lista de los tipos de datos (int, float, String), los campos nombre de la variable y valor. | |  |
| 6 |  | | Ingresa los datos solicitados y da clic en guardar se cierra el cuadro de diálogo. |
| 7 |  | | Clic en eliminar. |
| 8 | Muestra un cuadro de diálogo para confirmar la acción, en caso de aceptar se elimina y muestra el mensaje “**MSJ-08**” | |  |
|  |  | | Clic en ver parámetros. |
|  | Muestra los parámetros asociados a la salida seleccionada con la opción de eliminar. | |  |
|  |  | | Clic en eliminar (Parámetros). |
|  | Muestra un cuadro de diálogo para confirmar la acción, en caso de aceptar se elimina y muestra el mensaje “**MSJ-09**”. | |  |
|  | **Reglas de negocio** | | |
|  | -Es necesario cargar como mínimo un parámetro para cada caso de prueba.  -Las salidas deben de contener el mismo número de parámetros para el mismo ejercicio. | | |
|  | **Mensajes del sistema** | | |
|  | MSJ-08 Salida Eliminada.  MSJ-09 Parámetro eliminado. | | |

Tabla 10 SEA-006 Crear casos de prueba.

### 3.4.1.8 SEA-007-00 Inscribirse a una clase.

En la Tabla 11 se muestra el escenario que permite a los alumnos inscribirse a una clase antes creada por un profesor.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-007 Inscribirse a una clase. | |
| **Actores** | | Alumno y Sistema. | |
| **Descripción** | | Permite al alumno inscribirse a una clase. | |
| **Pre condiciones** | | Que el profesor anteriormente ya haya creado una clase.  Haber iniciado sesión como alumno. | |
| **Post condiciones** | | N/A. | |
| **Escenario 07 Inscribirse a una clase** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | Después de iniciar sesión el alumno se mostrará una pantalla que listará a los profesores, las clases y el botón registrar.  También una tabla que listará a las clases que se inscribió el alumno con los botones de eliminar y ver. | |  |
| 2 |  | | Selecciona un profesor de la lista. |
| 3 | Se actualiza el siguiente menú, lista las clases del profesor que anteriormente se ha seleccionado. | |  |
| 4 |  | | Selecciona una clase y da clic en registrar. |
| 5 | La tabla se actualizará y mostrará la clase a la que el alumno se ha inscrito con el botón de eliminar la clase y el de ver que enlazará a una nueva ventana dónde se mostrar el temario correspondiente a esa clase. | |  |
| 6 |  | | Clic en eliminar clase. |
| 7 | Muestra un cuadro de diálogo para confirmar la acción, en caso de aceptar se elimina y muestra el mensaje “**MSJ-10**” y actualiza la tabla. | |  |
| 8 |  | | Clic en “ver”. |
| 9 | El sistema direccionará a la siguiente pantalla “Temario” (alumno). | |  |
|  | **Reglas de negocio** | | |
|  | N/A. | | |
|  | **Mensajes del sistema** | | |
|  | MSJ-10 Clase Eliminada. | | |

Tabla 11 SEA-007 Inscribirse a una clase.

### 3.4.1.9 SEA-008-00 Resolver ejercicios

En la Tabla 12 se muestra el escenario que permite al alumno acceder al temario.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-008-00 Temario (Alumno). | |
| **Actores** | | Alumno y Sistema. | |
| **Descripción** | | Permite acceder al temario de una clase. | |
| **Pre condiciones** | | Que el profesor haya creado una clase. | |
| **Post condiciones** | | N/A. | |
| **Escenario 08 Temario (Alumno)** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | Muestra la página “Temario” con un TreeTable que listará los temas y subtemas de la clase. | |  |
| 2 |  | | Abrir un tema (nodo) para visualizar los subtemas. |
| 3 |  | | Clic en un subtema. |
| 4 | Muestra la página Ejercicios Alumno que listará los nombres de los ejercicios respecto al subtema seleccionado. | |  |
|  | **Reglas de negocio** | | |
|  | N/A. | | |
|  | **Mensajes del sistema** | | |
|  | N/A. | | |

Tabla 12 SEA-008-00 Temario (Alumno)

En la Tabla 13 se muestra el escenario que permite al alumno Resolver los ejercicios.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-008-00 Resolver Ejercicio. | |
| **Actores** | | Alumno y Sistema. | |
| **Descripción** | | Permite acceder y resolver un ejercicio. | |
| **Pre condiciones** | | Que el profesor haya creado un ejercicio. | |
| **Post condiciones** | | N/A. | |
| **Escenario 08 Resolver Ejercicios** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 |  | | Seleccionar un ejercicio de la pantalla alumno ejercicio. |
| 2 | Mostrará la página alumnosWorkSpace. Con los siguientes datos Nombre del ejercicio, descripción, pista, total de intentos, tiempo total, el cuerpo del programa, todo acorde a lo que anteriormente el profesor creo para este ejercicio. | |  |
| 3 | Al cargar la página el sistema deberá comprobar que el ejercicio exista en la base de datos de los alumnos, mostrar las modificaciones que se le han realizado al cuerpo del ejercicio. Iniciar un contador para los intentos y para el tiempo, una vez superados mostrará los mensajes “**MSJ-11**” y “**MSJ-12**” respectivamente. Y bloqueará el botón calificar. | |  |
| 4 |  | | El alumno aplicará la lógica que crea necesaria para resolver el ejercicio y deberá dar clic en el botón calificar. |
|  | El sistema evaluará respecto a las salidas y parámetros que el profesor ha creado anteriormente para este ejercicio y listará en una tabla las salidas aprobadas (OK) de lo contrario (FAIL). | |  |
|  |  | |  |
|  | **Reglas de negocio** | | |
|  | N/A. | | |
|  | **Mensajes del sistema** | | |
|  | MSJ-11 Número de intentos superados.  MSJ-12 Tiempo superado. | | |

Tabla 13 SEA-008-00 Resolver Ejercicio.

### 3.4.1.10 SEA-009-00 Administrar avances

En la Tabla 14 se muestra el escenario que permite al profesor visualizar las gráficas y reporte de avance de sus alumnos.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-009-00 Visualizar gráficas y reportes (Profesor). | |
| **Actores** | | Profesor y Sistema. | |
| **Descripción** | | Permite visualizar gráficas y reportes de avances de los alumnos inscritos a la clase. | |
| **Pre condiciones** | | Que los alumnos hayan resuelto algún ejercicio. | |
| **Post condiciones** | | N/A. | |
| **Escenario 09 Visualizar gráficas (Profesor)** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | La opción se encuentra el Menú y solo se mostrará una vez que se ha entrado a una clase. | |  |
| 2 |  | | Clic en la opción del menú Mis gráficas |
|  | Con una gráfica general con el estado actual de los ejercicios que pertenecen a esa clase; también mostrará una tabla con los alumnos inscritos y las opciones de mostrar los gráficas individuales (Número de intentos, Tiempo empleado en cada ejercicio) y reportes. | |  |
| 4 |  | | Clic en ver reporte. |
| 5 | Mostrará un cuadro de diálogo con las opciones de ver reporte por clase, tema, subtema, ejercicio y alumno. Al seleccionar alguna opción mostrará el reporte correspondiente. | |  |
|  | También se podrá visualizar reportes generales la clase, tema, subtema y ejercicio, el cual listará todos los alumnos que correspondan. | |  |
|  | **Reglas de negocio** | | |
|  | N/A. | | |
|  | **Mensajes del sistema** | | |
|  | N/A. | | |

Tabla 14 SEA-009-00 Visualizar gráficas y reportes (Profesor).

En la Tabla 15 se muestra el escenario que permite al Alumno visualizar las gráficas de sus avances.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-009-00 Visualizar gráficas y reportes (Alumno). | |
| **Actores** | | Alumno y Sistema. | |
| **Descripción** | | Permite visualizar al alumno sus gráficas y reportes de avances. | |
| **Pre condiciones** | | Que los alumnos hayan resuelto algún ejercicio. | |
| **Post condiciones** | | N/A. | |
| **Escenario 09 Visualizar gráficas (Alumno)** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | La opción se encuentra el Menú y solo se mostrará una vez que se ha entrado a una clase. | |  |
| 2 |  | | Clic en la opción del menú Mis gráficas. |
| 3 | Cargará la página, las gráficas de avances con las siguientes categorías: Número de intentos, Tiempo empleado en cada ejercicio, Días empleados .botones de Ver Reporte. | |  |
| 4 |  | | Clic en ver reporte. |
| 5 | Mostrará un cuadro de diálogo con las opciones de ver reporte por clase, tema, subtema y ejercicio. Al seleccionar alguna opción mostrará el reporte correspondiente. | |  |
|  | **Reglas de negocio** | | |
|  | N/A. | | |
|  | **Mensajes del sistema** | | |
|  | N/A. | | |

Tabla 15 SEA-009-00 Visualizar gráficas y reportes (Alumno).

En la Tabla 16 se muestra el escenario que permite al Alumno visualizar un reporte en PDF de sus avances.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-009-00 Reporte PDF Alumno. | |
| **Actores** | | Alumno y Sistema. | |
| **Descripción** | | Permite visualizar al alumno un reporte de avances en formato PDF. | |
| **Pre condiciones** | | Que los alumnos hayan resuelto algún ejercicio. | |
| **Post condiciones** | | N/A. | |
| **Escenario 09 Visualizar gráficas (Alumno)** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | La opción para visualizar el Reporte se localiza en la misma página que el alumno consulta sus gráficas. | |  |
| 2 |  | | Clic en la opción “Ver Reporte”. |
| 3 | Inmediatamente el sistema creará un reporte con los siguientes datos: nombre del alumno, clase, tema, ejercicio, fecha de inicio y terminación, intentos realizados, minutos que se han invertido y si el alumno ha pasado. | |  |
|  | **Reglas de negocio** | | |
|  | N/A. | | |
|  | **Mensajes del sistema** | | |
|  | N/A. | | |

Tabla 16 SEA-009-00 Reporte PDF Alumno.

En la Tabla 17 se muestra el escenario que permite al Profesor visualizar un reporte en PDF de sus avances.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | | SEA-009-00 Reporte PDF Alumno. | |
| **Actores** | | Profesor y Sistema. | |
| **Descripción** | | Permite visualizar al alumno un reporte de avances en formato PDF. | |
| **Pre condiciones** | | Que los alumnos hayan resuelto algún ejercicio. | |
| **Post condiciones** | | N/A. | |
| **Escenario 09 Visualizar gráficas (Profesor)** | | | |
| **No.** | **Respuesta del sistema** | | **Acción del actor** |
| 1 | La opción para visualizar el Reporte se localiza en la misma página que el alumno consulta sus gráficas. | |  |
| 2 |  | | Clic en la opción “Ver Reporte”. |
|  | El sistema lanzará un cuadro de diálogo con las siguientes categorías: por Clase, Tema, Subtema, Ejercicio y alumno. | |  |
|  |  | | Clic en alguna de las opciones. |
|  | Mostrará un segundo cuadro de diálogo listando el campo solicitado con un botón de ver reporte. | |  |
|  |  | | Clic en ver reporte. |
| 3 | Inmediatamente el sistema creará un reporte con los siguientes datos: nombre del alumno, clase, tema, ejercicio, fecha de inicio y terminación, intentos realizados, minutos que se han invertido y si el alumno ha pasado, de acuerdo a la categoría que seleccionado. | |  |
|  | **Reglas de negocio** | | |
|  | N/A. | | |
|  | **Mensajes del sistema** | | |
|  | N/A. | | |

Tabla 17 SEA-009-00 Reporte PDF Alumno

### Diagramas de clases

El diagrama de clases es un modelo estático que describe las clases existentes, los atributos y el comportamiento que tiene, además es un diagrama que captura la estructura lógica del sistema.

Los diagramas de clases son los más utilizados para ilustrar las relaciones de clases y sus interfaces. Las generalidades, las agregaciones y las asociaciones son todas importantes para reflejar la herencia.

A continuación se visualizarán los diagramas de clases de cada caso de uso existente sirviendo de sus características para tener una perspectiva a nivel de estructura de clases y atributos de los cuales se comprende el sistema.

### Diagramas de clases

En la Figura 13 se muestra el diagrama de clases referente al caso de usoSEA-CU-001-00 Crear Cuenta.

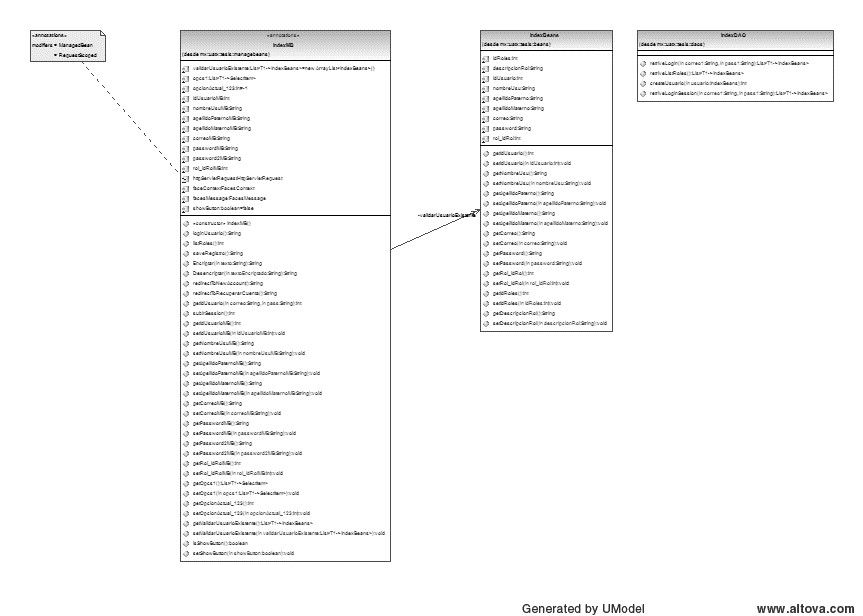


Figura 13 Diagrama de clases SEA-CU-001-00 Crear Cuenta

En la Figura 14 se muestra el diagrama de clases referente al caso de usoSEA-CU-002-00 Recuperar Cuenta.

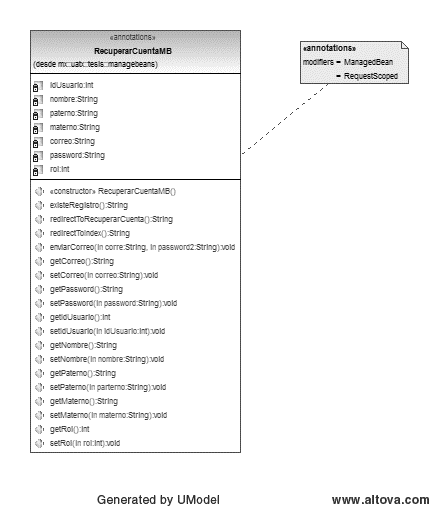
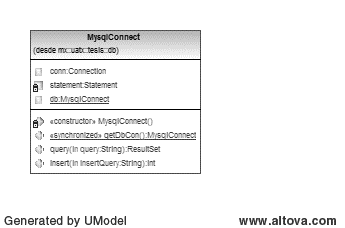
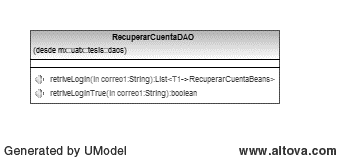
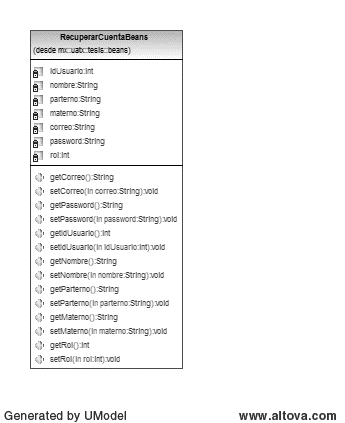


Figura 14 Diagrama de clases SEA-CU-002-00 Recuperar Cuenta.

En la Figura 15 se muestra el diagrama de clases referente al caso de usoSEA-CU-003-00 Iniciar Sesión.

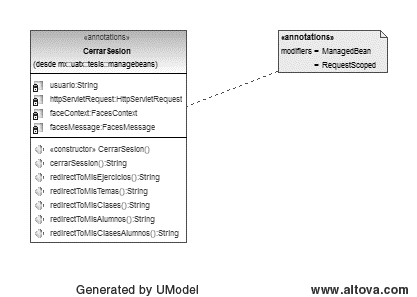
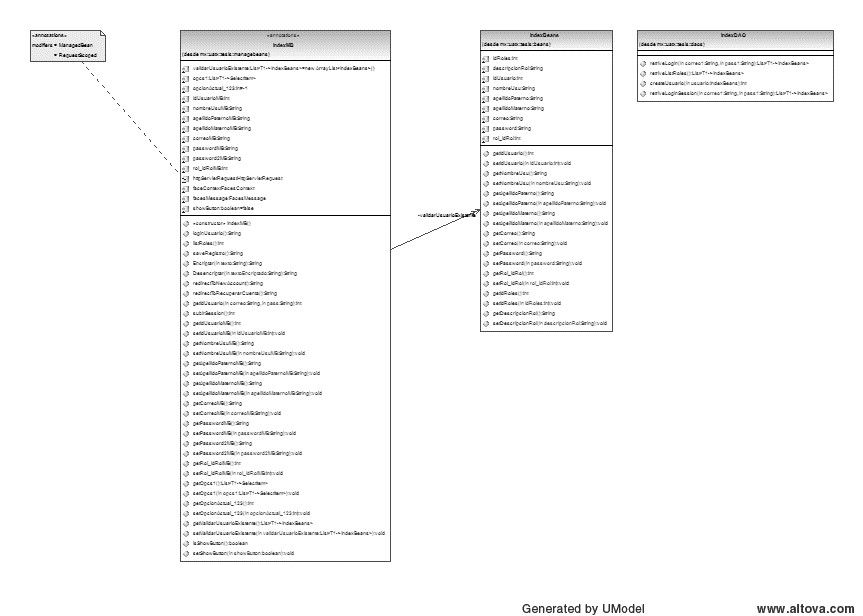


Figura 15 Diagrama de clases SEA-CU-003-00 Iniciar Sesión.

En la Figura 16 se muestra el diagrama de clases referente al caso de usoSEA-CU-004-00 Administrar clases.

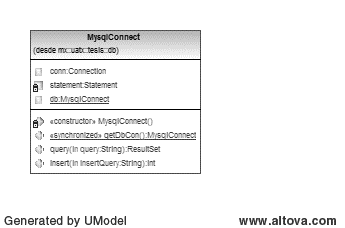
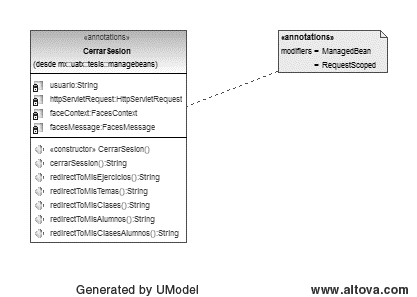
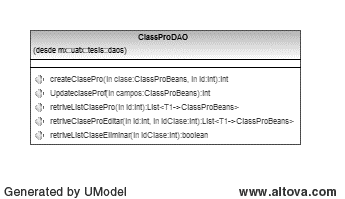
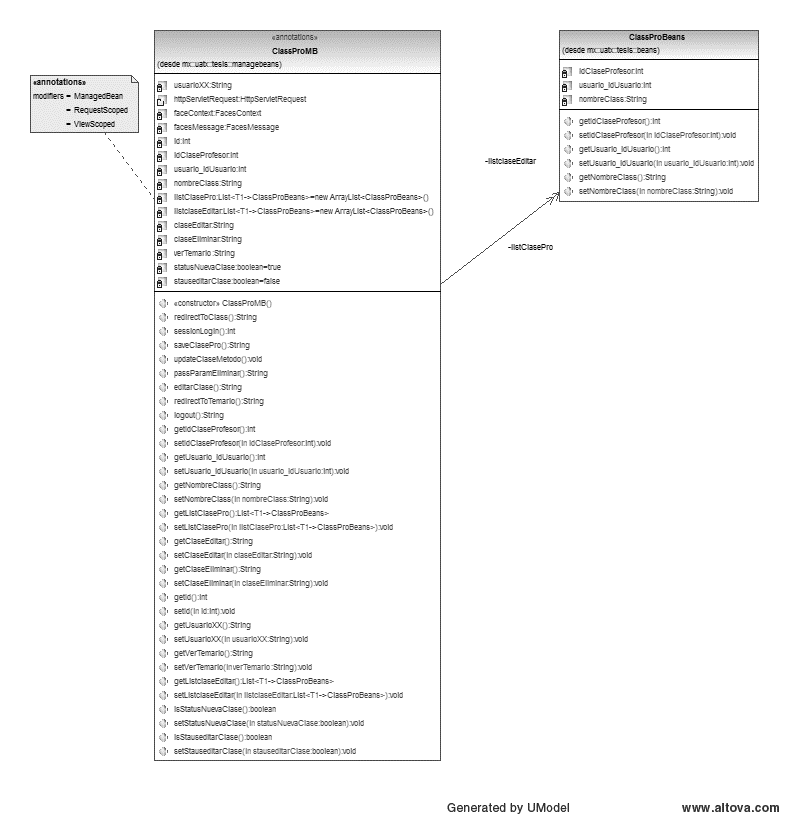


Figura 16 Diagrama de clases SEA-CU-005-00 Administrar clases.

En la Figura 17 se muestra el diagrama de clases referente al caso de uso SEA-CU-005-00 Administrar temario.

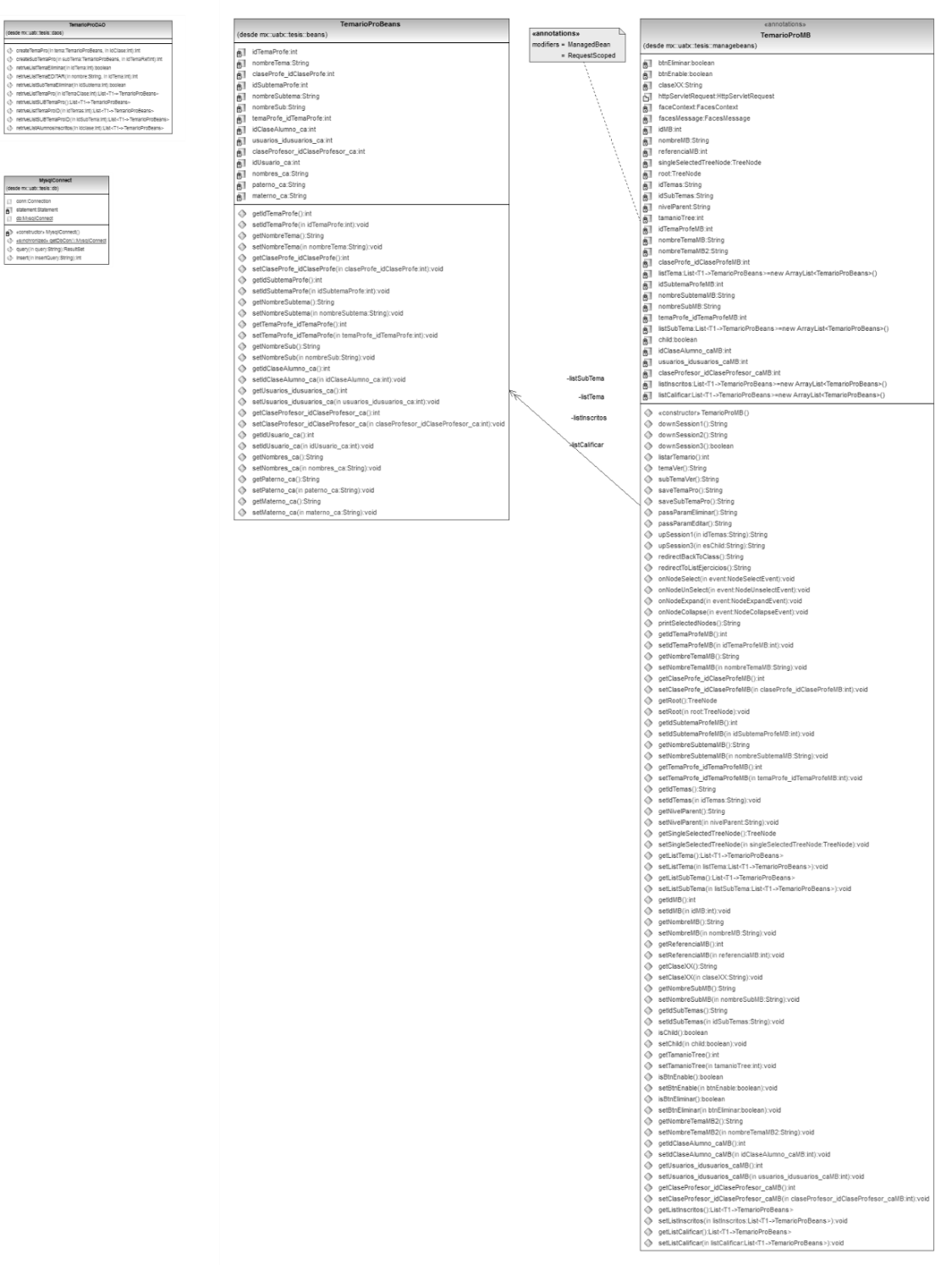
****

Figura 17 Diagrama de clases SEA-CU-005-00 Administrar temario.

Figura 3.1 Diagrama de clases SEA-CU-005-00 Administrar temario.

Figura 4. Diagrama de clases SEA-CU-006-00 Administrar ejercicios.Figura 3.2 Diagrama de clases SEA-CU-005-00 Administrar temario.

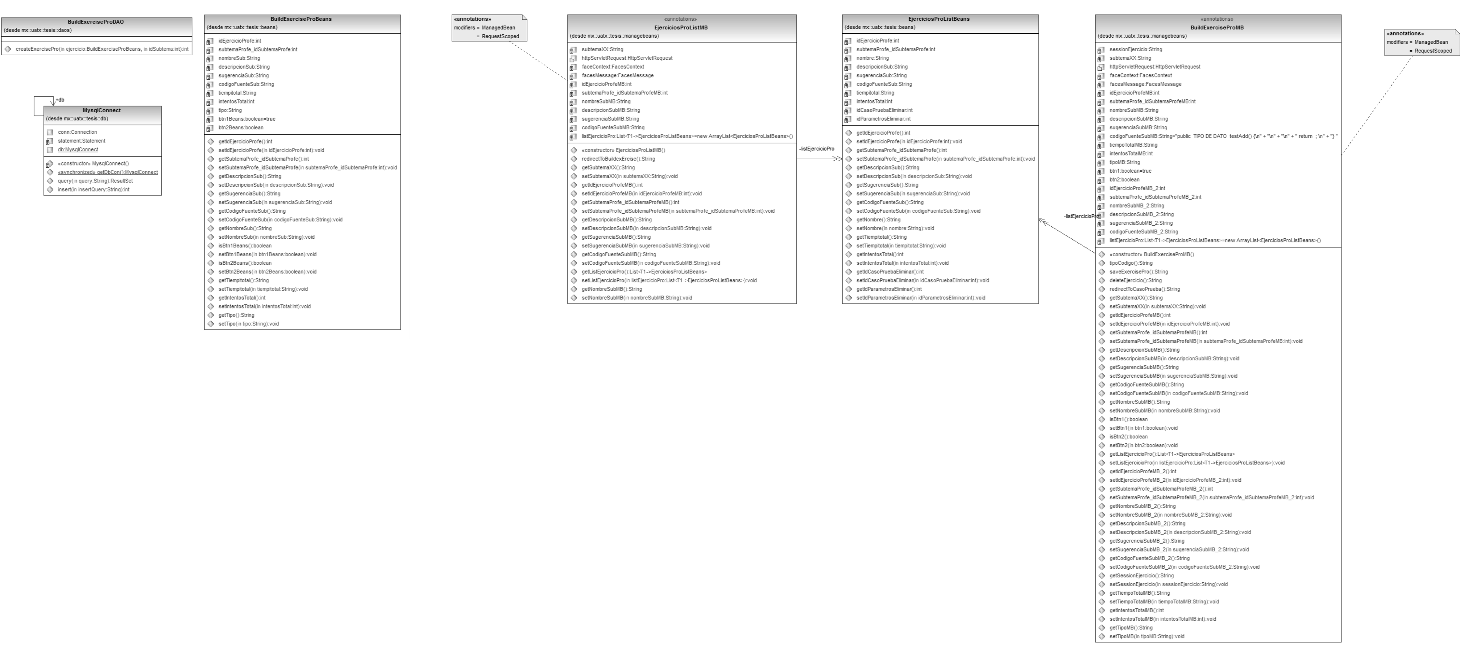
En la **Figura 18** se muestra el diagrama de clases referente al caso de usoSEA-CU-006-00 Administrar ejercicios.

Figura 18 Diagrama de clases SEA-CU-006-00 Administrar ejercicios.

En la Figura 19 se muestra el diagrama de clases referente al caso de usoSEA-CU-007-00 Inscribirse a una clase.

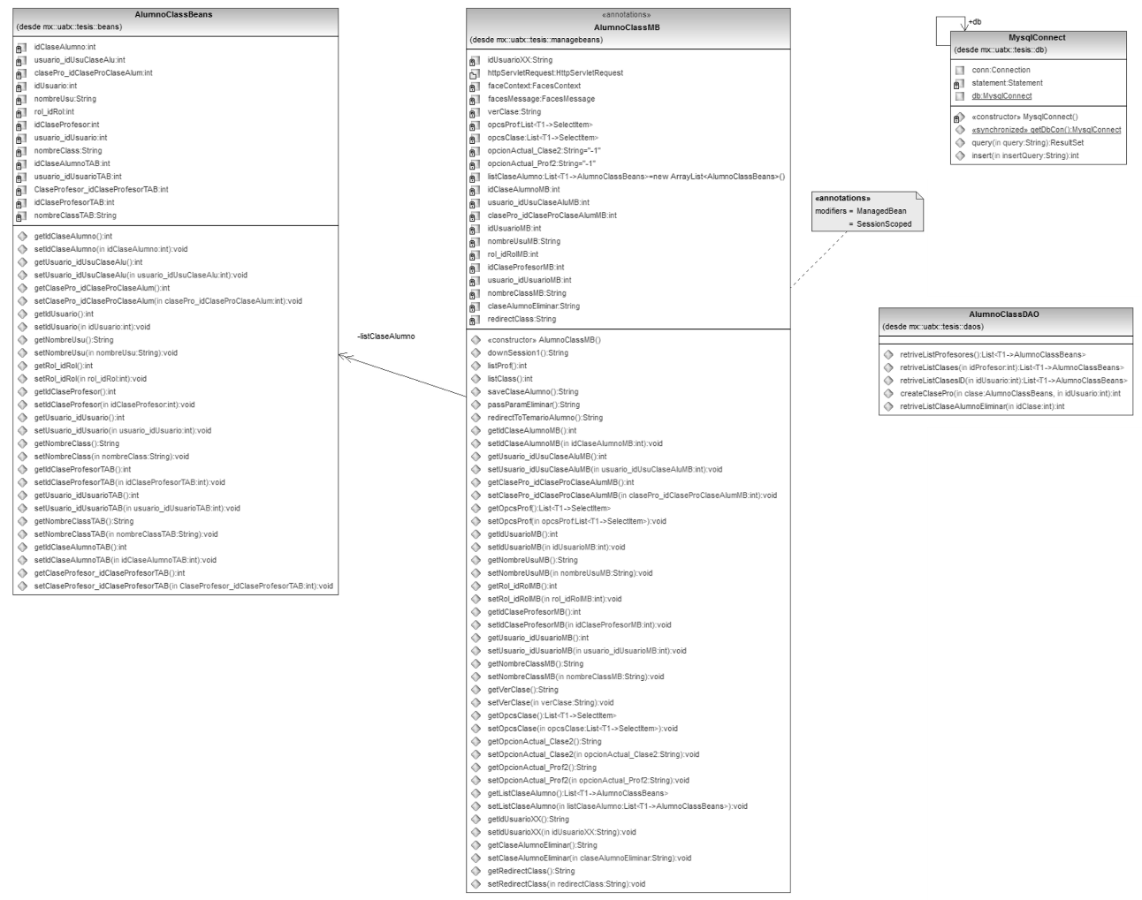
****

Figura 19 Diagrama de clases SEA-CU-007-00 Inscribirse a una clase.

En la Figura 20 se muestra el diagrama de clases referente al caso de usoSEA-CU-008-00 Resolver Ejercicio.

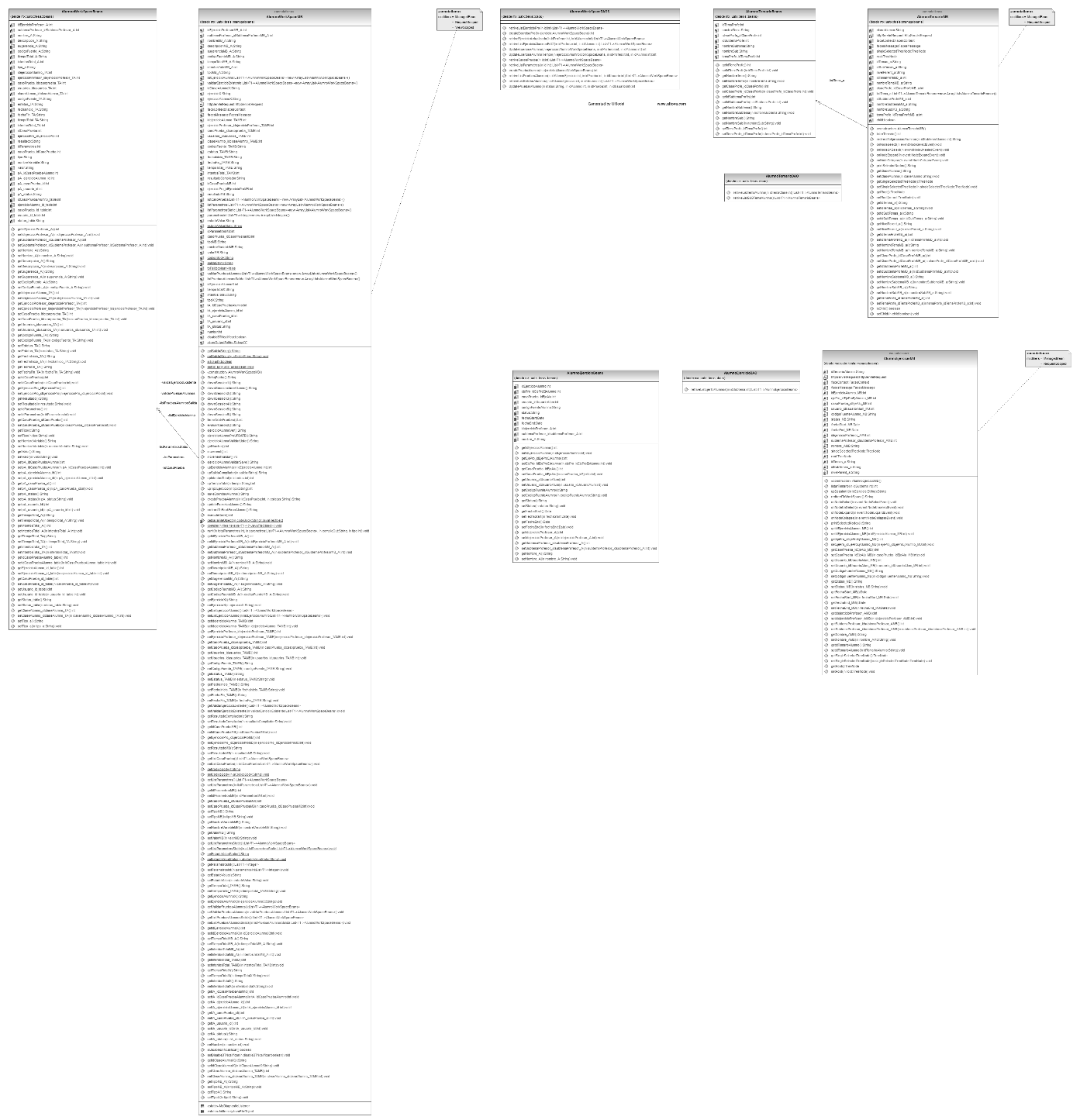
****

Figura 20 Diagrama de clases SEA-CU-008-00 Resolver Ejercicio.

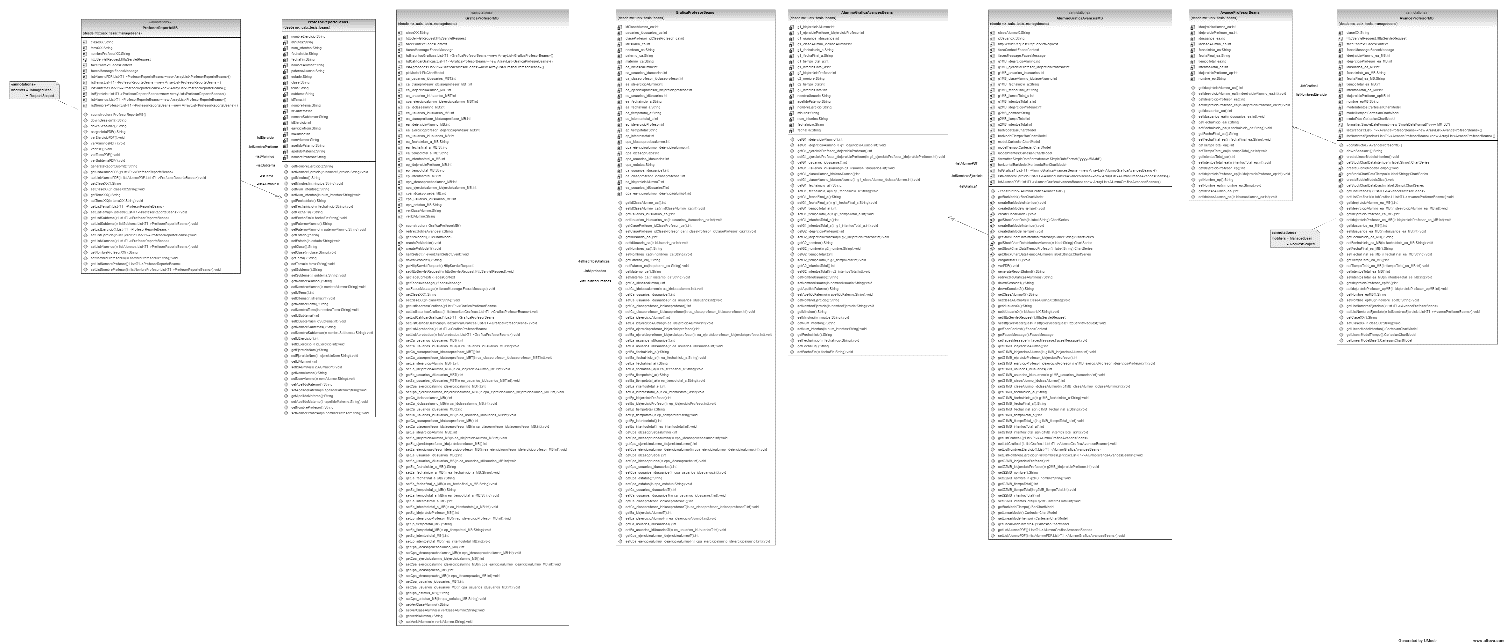
En la Figura 21 se muestra el diagrama de clases referente al caso de usoSEA-CU-009-00 Visualizar Gráficas.

Figura 21 Diagrama de clases uso SEA-CU-009-00 Visualizar Gráficas.

### Diagramas de secuencia

Los diagramas de secuencia muestran el intercambio de mensajes, es decir, la forma en que se invocan en un momento dado. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.

En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros.

En la Figura 22 se muestra el diagrama de secuencias referente al caso de uso SEA-CU-001-00 Crear Cuenta.

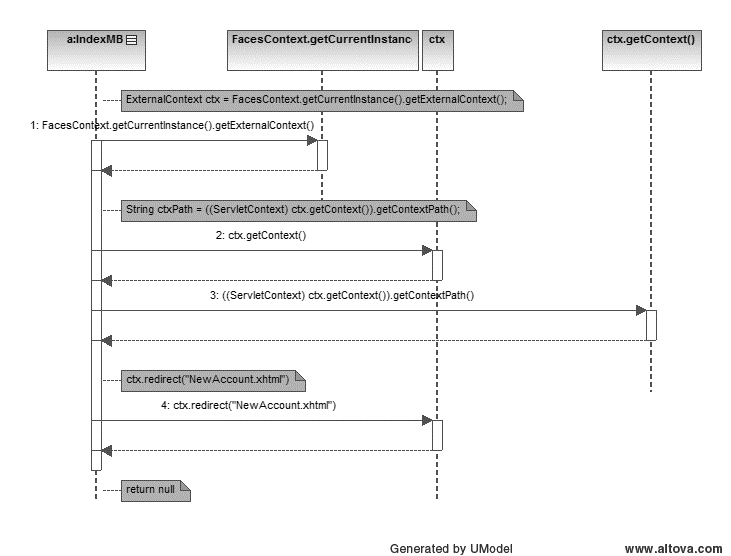


Figura 22 Diagrama de secuencias SEA-CU-001-00 Crear Cuenta.

En la Figura 23 se muestra el diagrama de secuencias referente al caso de uso SEA-CU-002-00 Recuperar Cuenta.

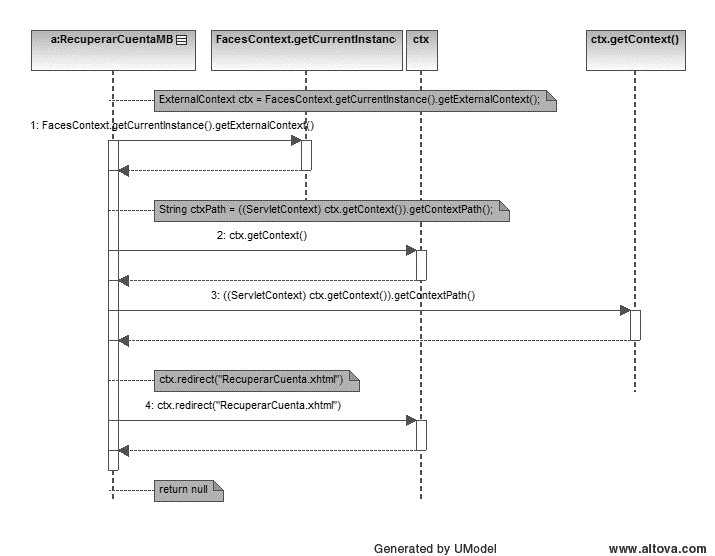


Figura 23 Diagrama de secuencias SEA-CU-002-00 Recuperar Cuenta.

En la Figura 24 se muestra el diagrama de secuencias referente al caso de uso SEA-CU-003-00 Iniciar Sesión

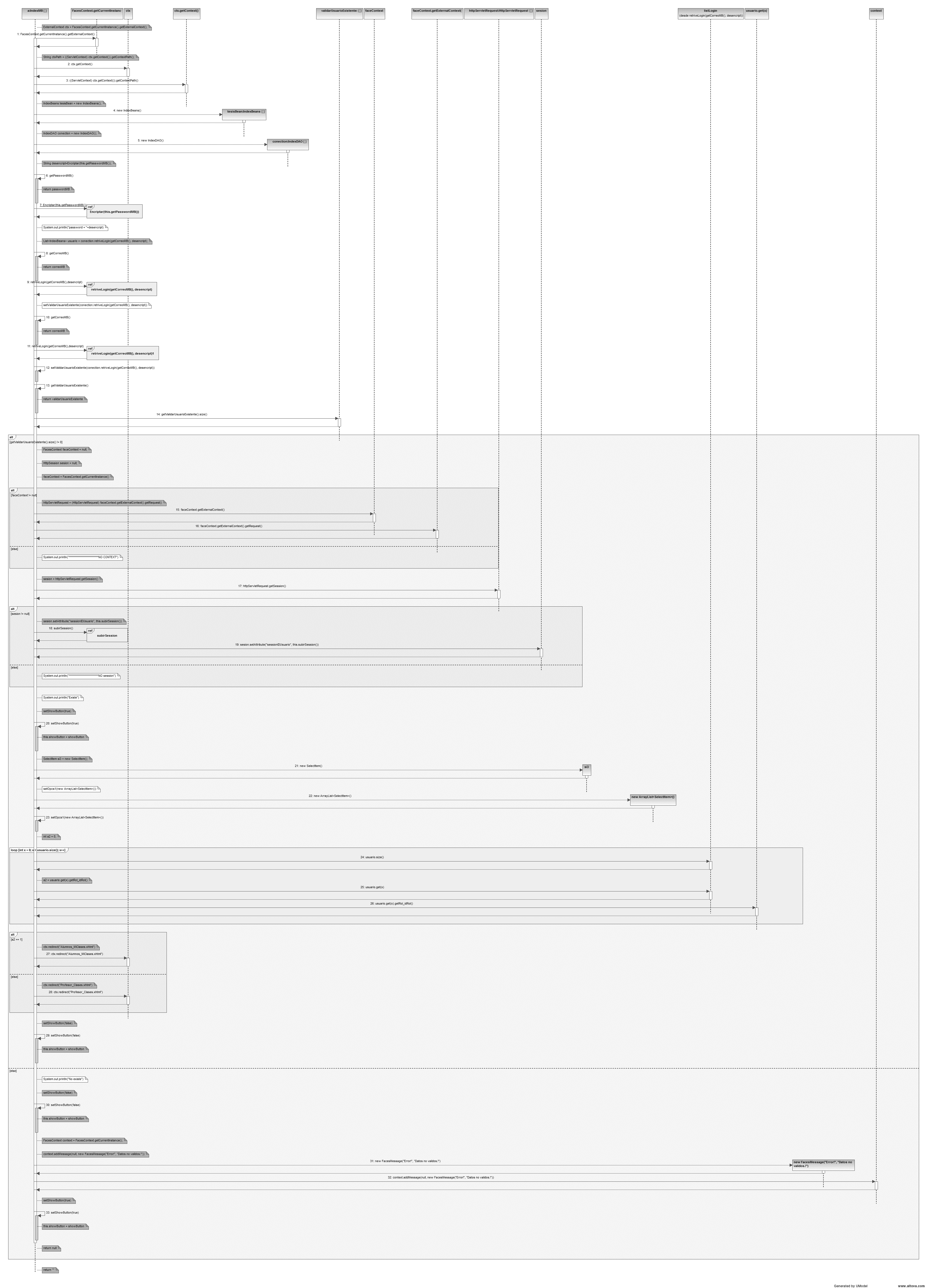


Figura 24 Diagrama de secuencias SEA-CU-003-00 Iniciar Sesión

En la Figura 25 se muestra el diagrama de secuencias referente al caso de uso SEA-CU-004-00 Administrar clases (Guardar una clase)

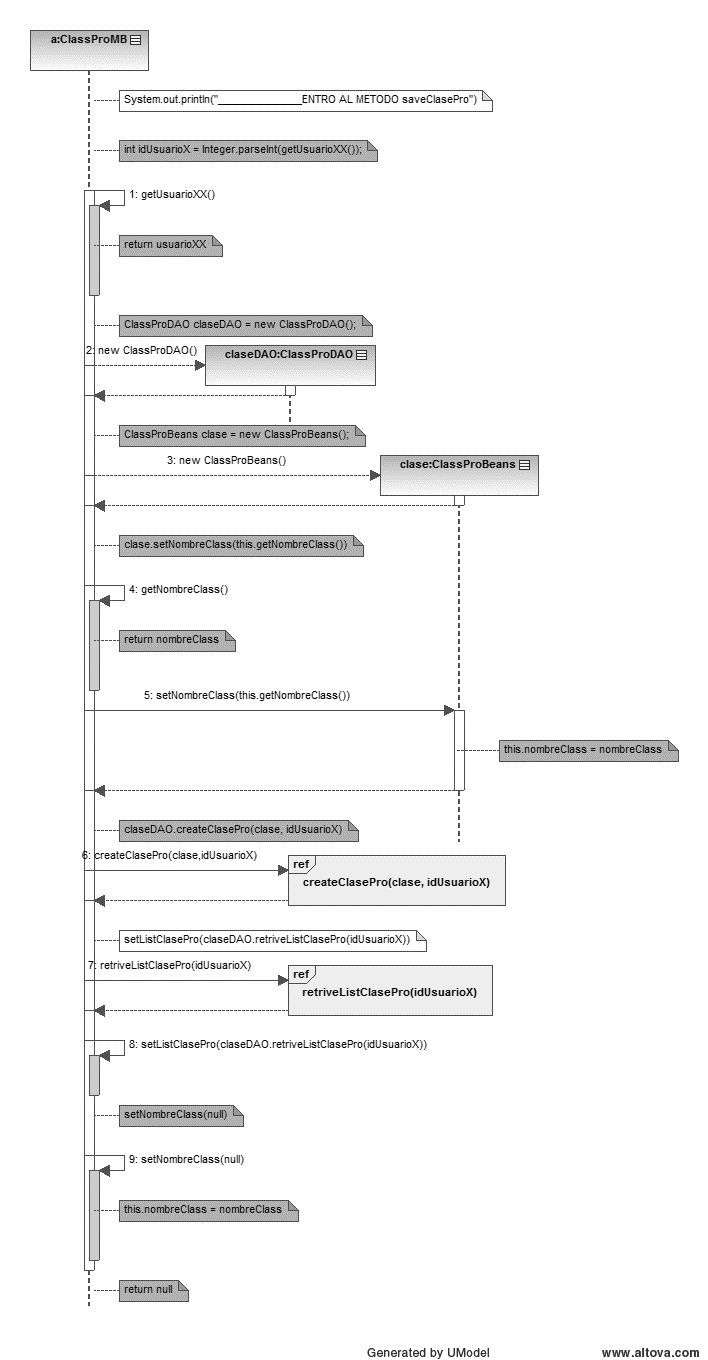


Figura 25 Diagrama de secuencias SEA-CU-004-00 Administrar clases

En la Figura 26 se muestra el diagrama de secuencias referente al caso de uso SEA-CU-004-00 Administrar clases (Eliminar una clase).

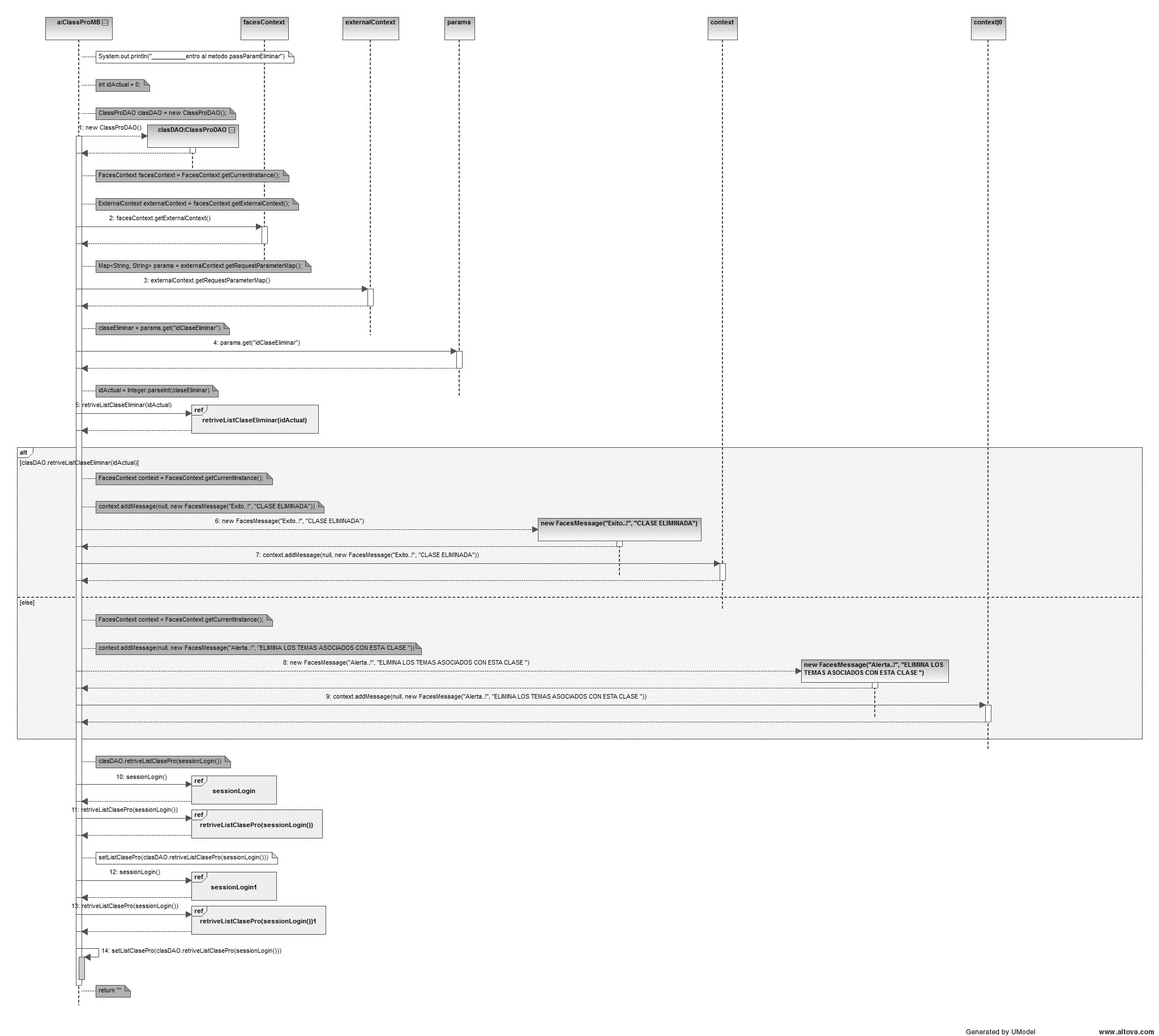


Figura 26 Diagrama de secuencias SEA-CU-004-00 Administrar clases.

En la Figura 27 se muestra el diagrama de secuencias referente al caso de uso SEA-CU-004-00 Administrar clases (Editar una clase)

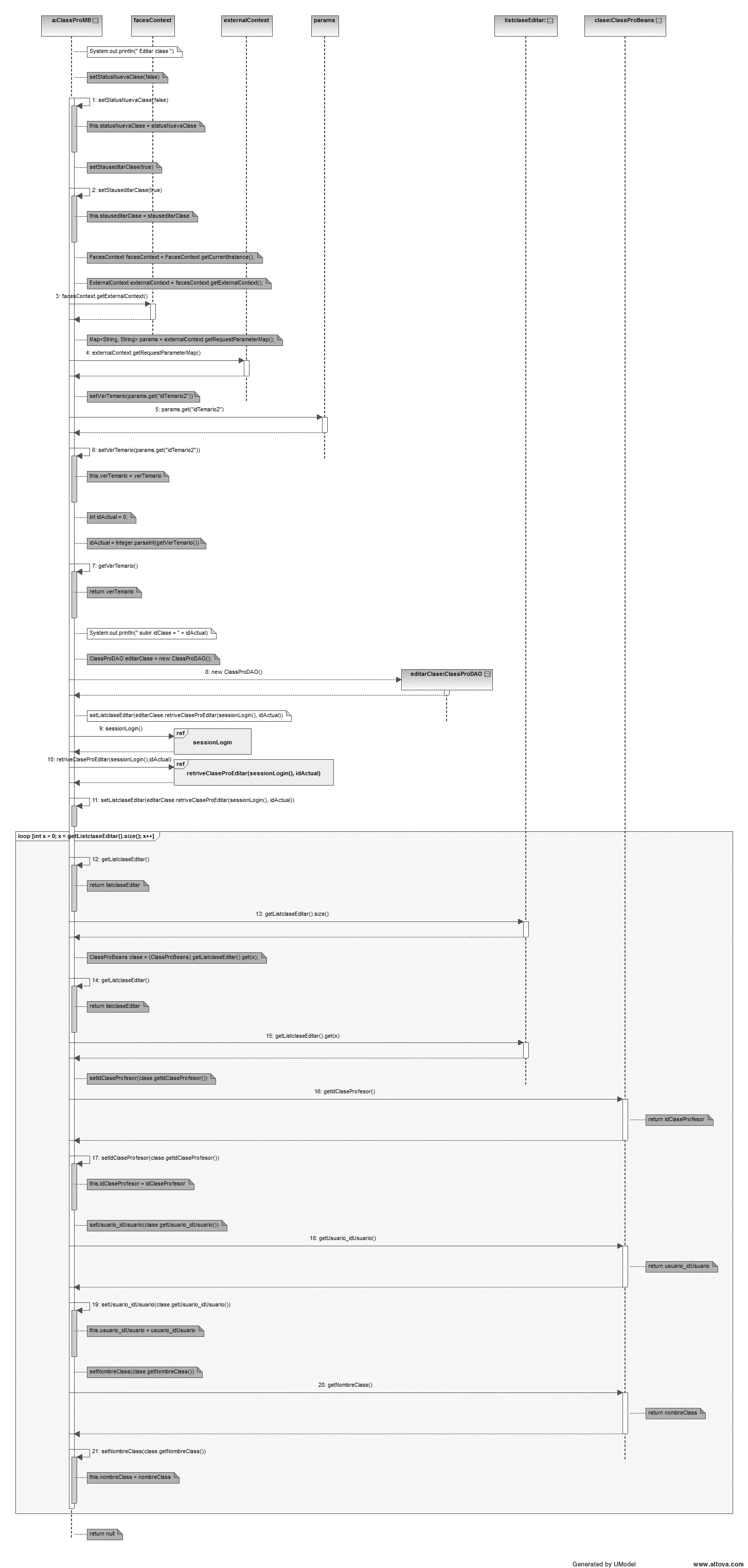


Figura 27 Diagrama de secuencias SEA-CU-004-00 Administrar clases.

En la Figura 28 se muestra el diagrama de secuencias referente al caso de uso SEA-CU-005-00 Administrar temario.

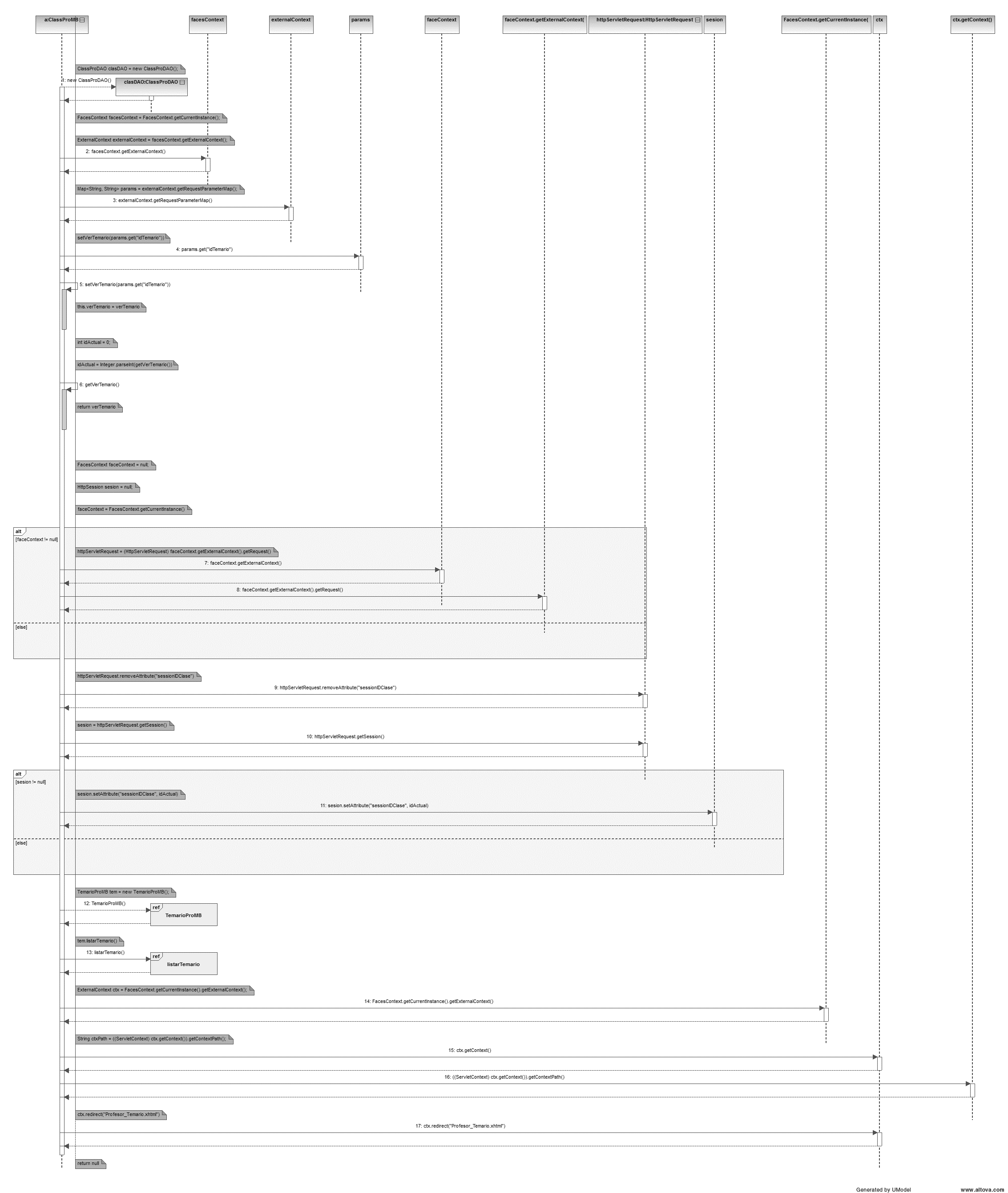


Figura 28 Diagrama de secuencias SEA-CU-005-00 Administrar temario.

Figura 3.3 Diagrama de secuencias SEA-CU-001-00 Crear Cuenta.

En la Figura 29 se muestra el diagrama de secuencias referente al caso de uso SEA-CU-006-00 Administrar ejercicios.

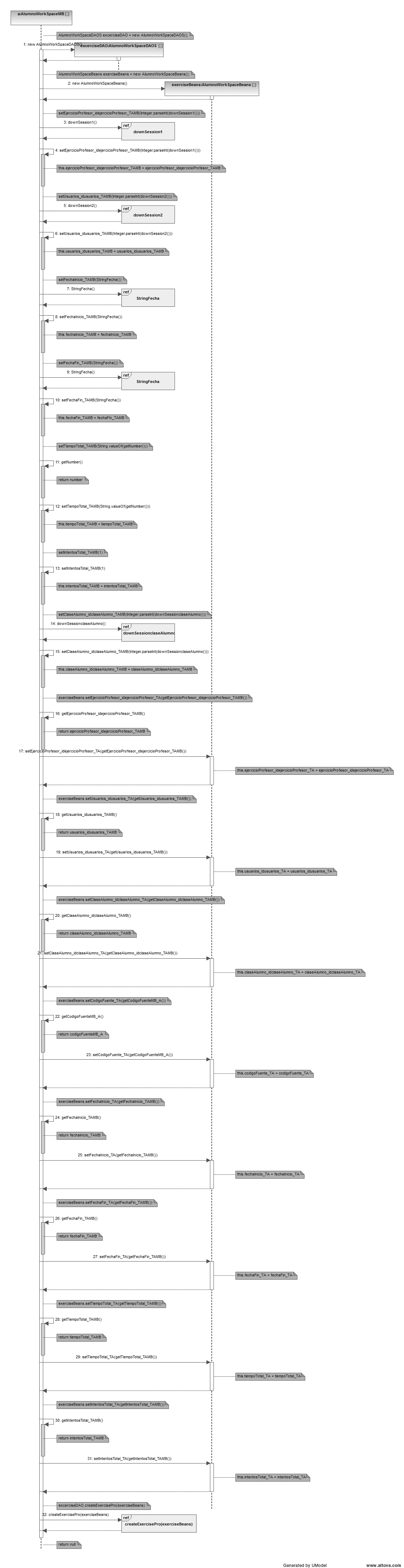


Figura 29 Diagrama de secuencias SEA-CU-006-00 Administrar ejercicios.

En la Figura 30 se muestra el diagrama de secuencias referente al caso de uso SEA-CU-007-00 Inscribirse a una clase.

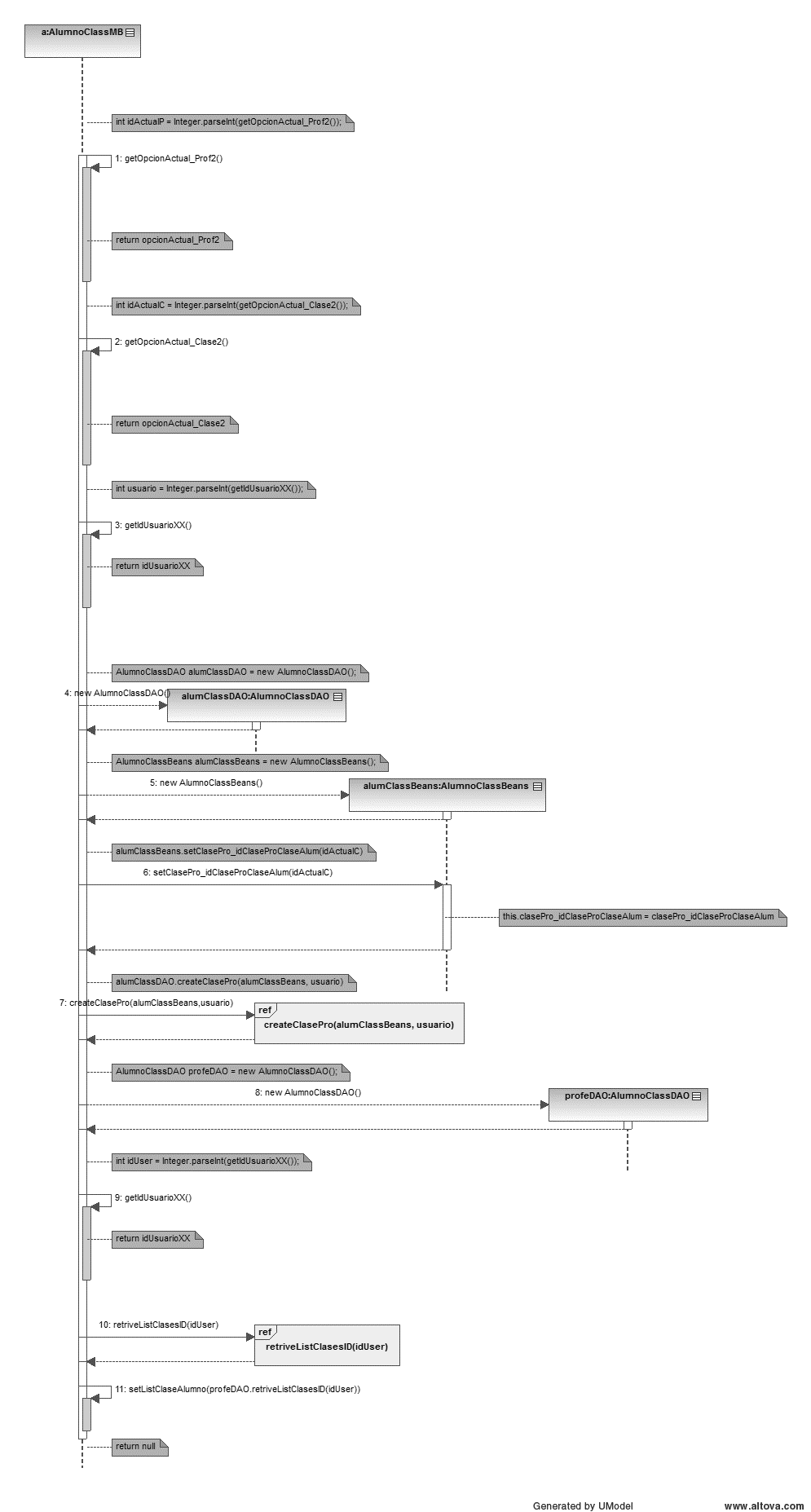


Figura 30 Diagrama de secuencias SEA-CU-007-00 Inscribirse a una clase.

En la Figura 31 se muestra el diagrama de secuencias referente al caso de uso SEA-CU-008-00 Resolver Ejercicio.

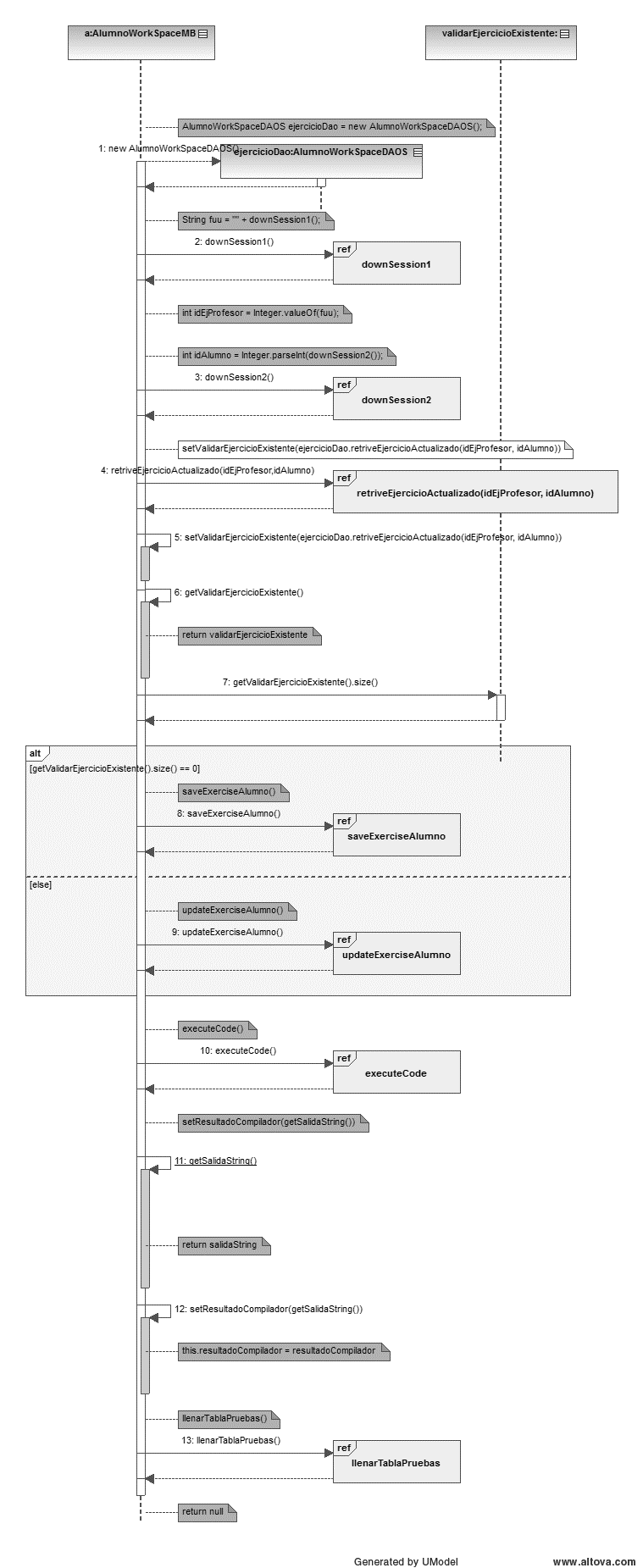


Figura 31 Diagrama de secuencias SEA-CU-008-00 Resolver Ejercicio.

En la Figura 32 se muestra el diagrama de secuencias referente al caso de uso SEA-CU-009-00 Visualizar Gráficas.

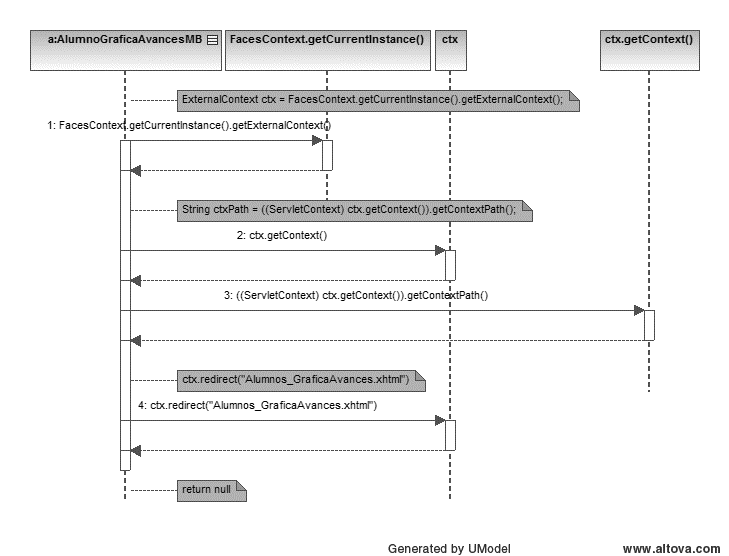


Figura 32 Diagrama de secuencias SEA-CU-009-00 Visualizar Gráficas.

### Diagrama de componentes

El diagrama de componentes describe los elementos físicos de un sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Todo tipo de elementos software que entran en la fabricación de aplicaciones informáticas, pueden ser simples archivos de datos, una tabla, un ejecutable, librerías, documentos digitales, entre otros. A continuación en la Figura 33 se muestra el diagrama de componentes del sistema SISE.

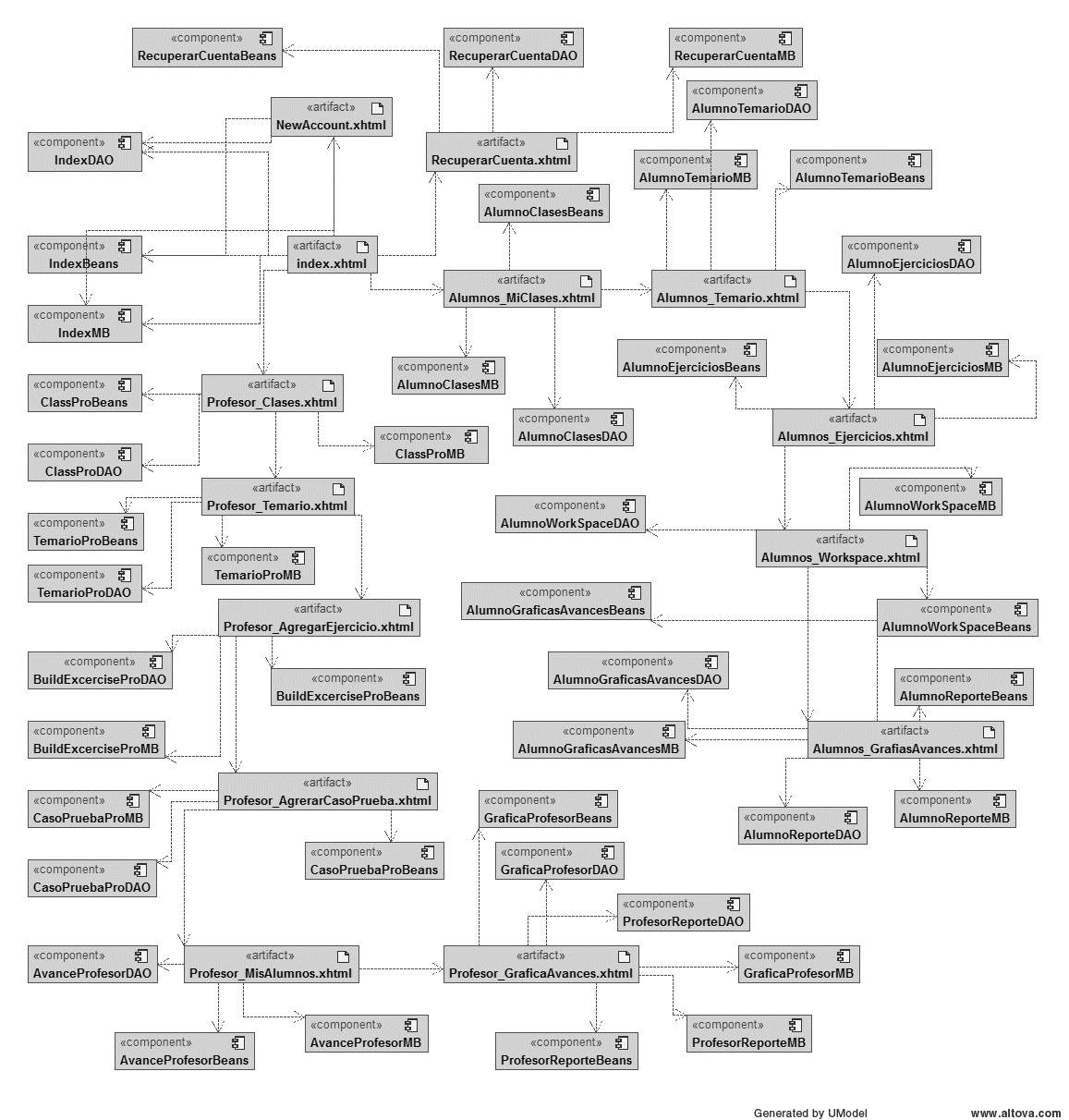


Figura 33 Diagrama de componentes SISE.

### Diagrama de despliegue

El diagrama de despliegue muestra la capa física del sistema, dando así un complemento del diagrama de componentes. El diagrama de despliegue muestra la configuración en funcionamiento del sistema incluyendo su software y hardware. Para cada componente de un diagrama es necesario que de deba documentar las características técnicas requeridas, el tráfico de la red, el tiempo de respuesta.

Se utiliza para describir la vista de despliegue estática de un sistema, relacionados con el diagrama de componentes mediante nodos conectados.

En la Figura 34 se muestra el diagrama de despliegue del sistema SEA.

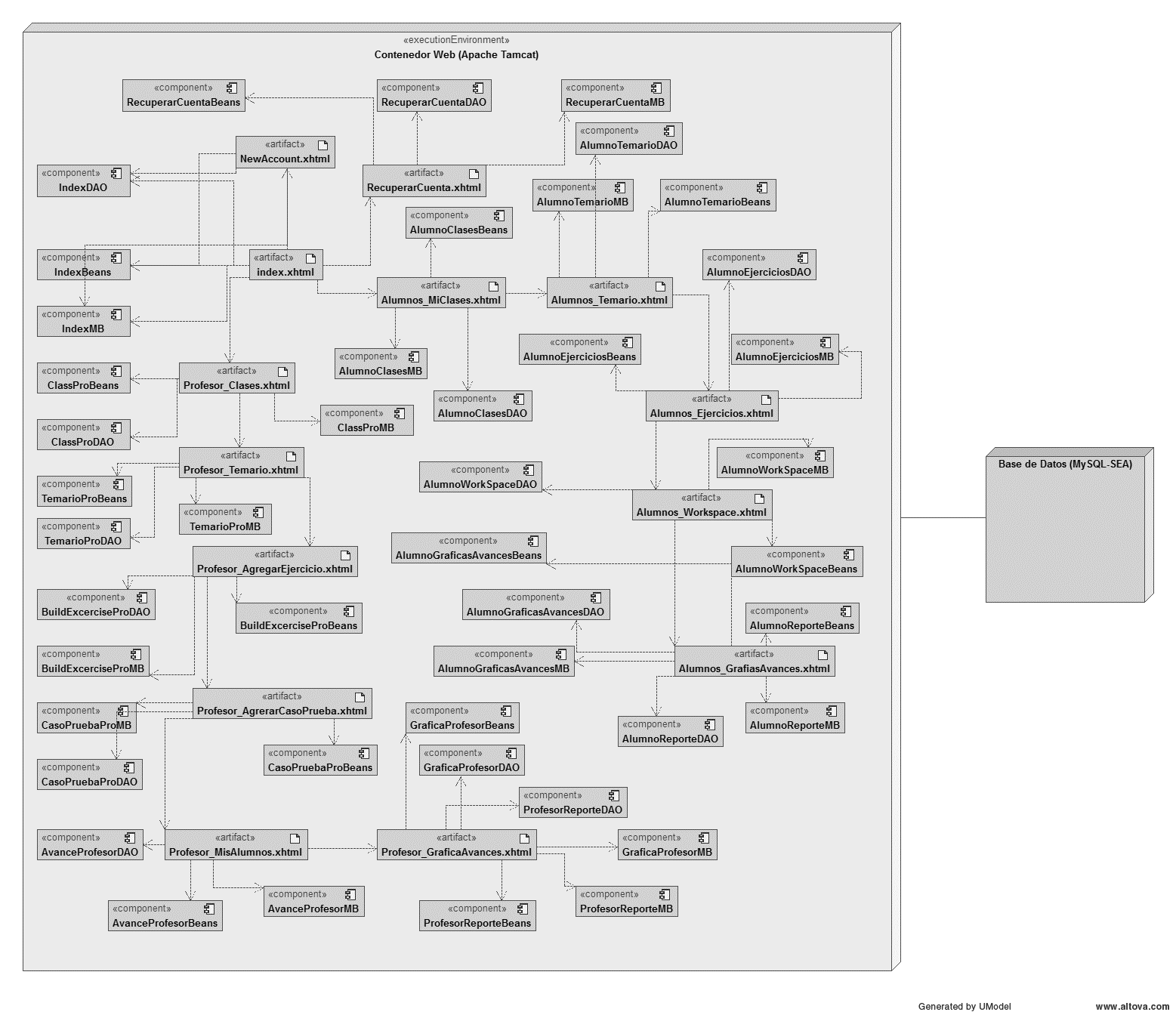


Figura 34 Diagrama de secuencias SEA-CU-001-00 Crear Cuenta.

## Interfaz gráfica del sistema SEA

En la siguiente información se mostrará la interfaz gráfica del sistema, cada imagen será presentada según los casas de uso y escenarios del sistema existente.

Para comenzar con el muestre de imágenes, cabe aclarar que al ejecutar el proyecto SEA se despliega la siguiente interfaz por defecto que se aprecia en la Figura 35



Figura 35 interfaz del inicio del sistema SEA

### SEA- CU-001-00 Crear cuenta

En la Figura 36 se muestra la interfaz de usuario del sistema para el registro de alumnos y profesores.

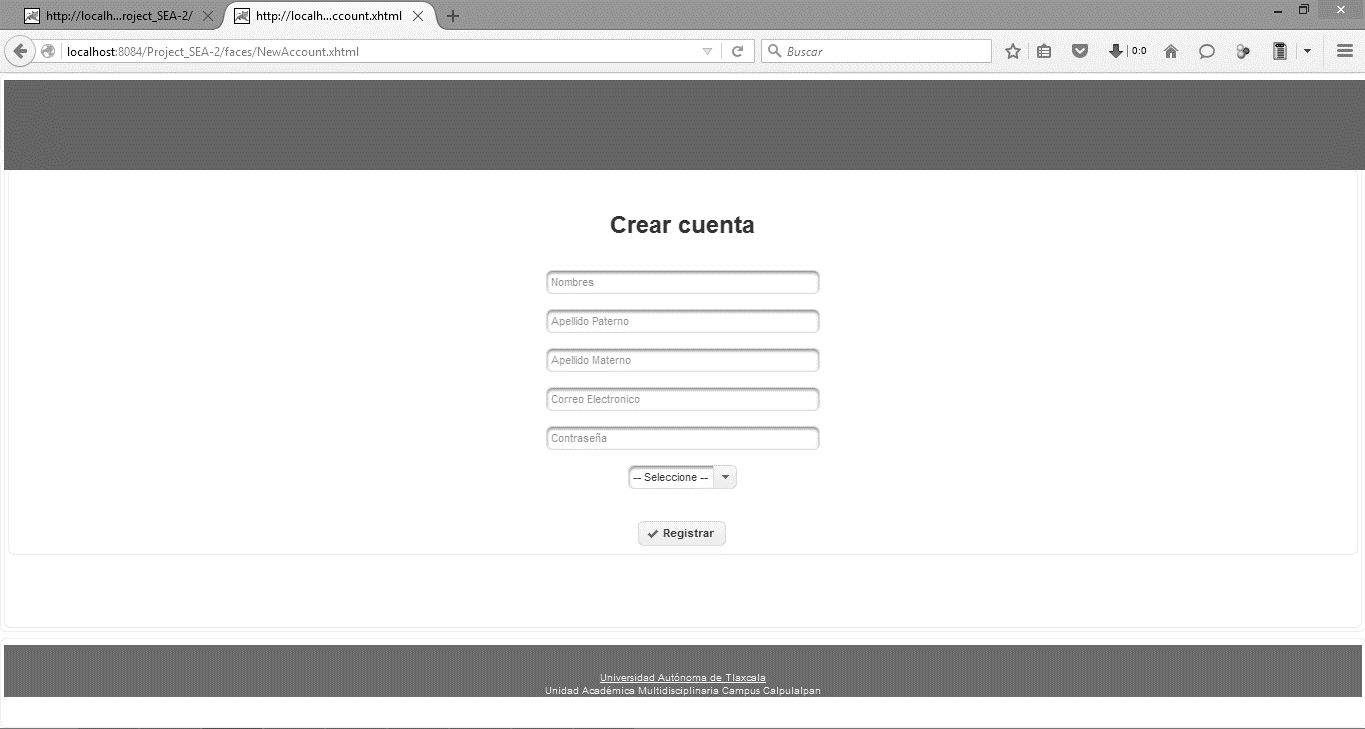


Figura 36 Interfaz de usuario SEA-CU-001-00 Crear Cuenta.

### SEA- CU-002-00 Recuperar clave de acceso.

En la Figura 37 se muestra la interfaz de usuario del sistema para recuperar la clave de acceso.

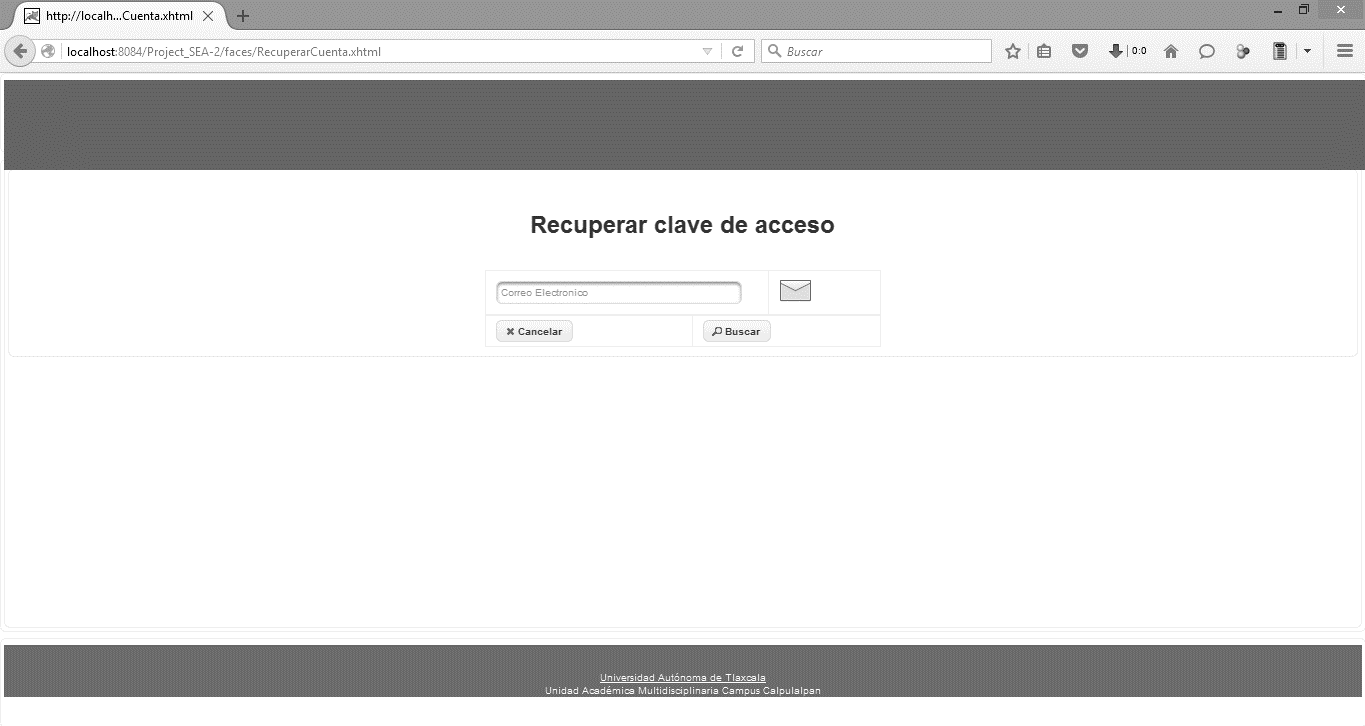


Figura 37 Interfaz de usuario SEA-002-00 Recuperar clave de acceso.

### SEA- CU-003-00 Administrar sesiones.

En la Figura 38 se muestra la interfaz dónde el profesor o alumno podrán iniciar sesión para acceder al sistema.

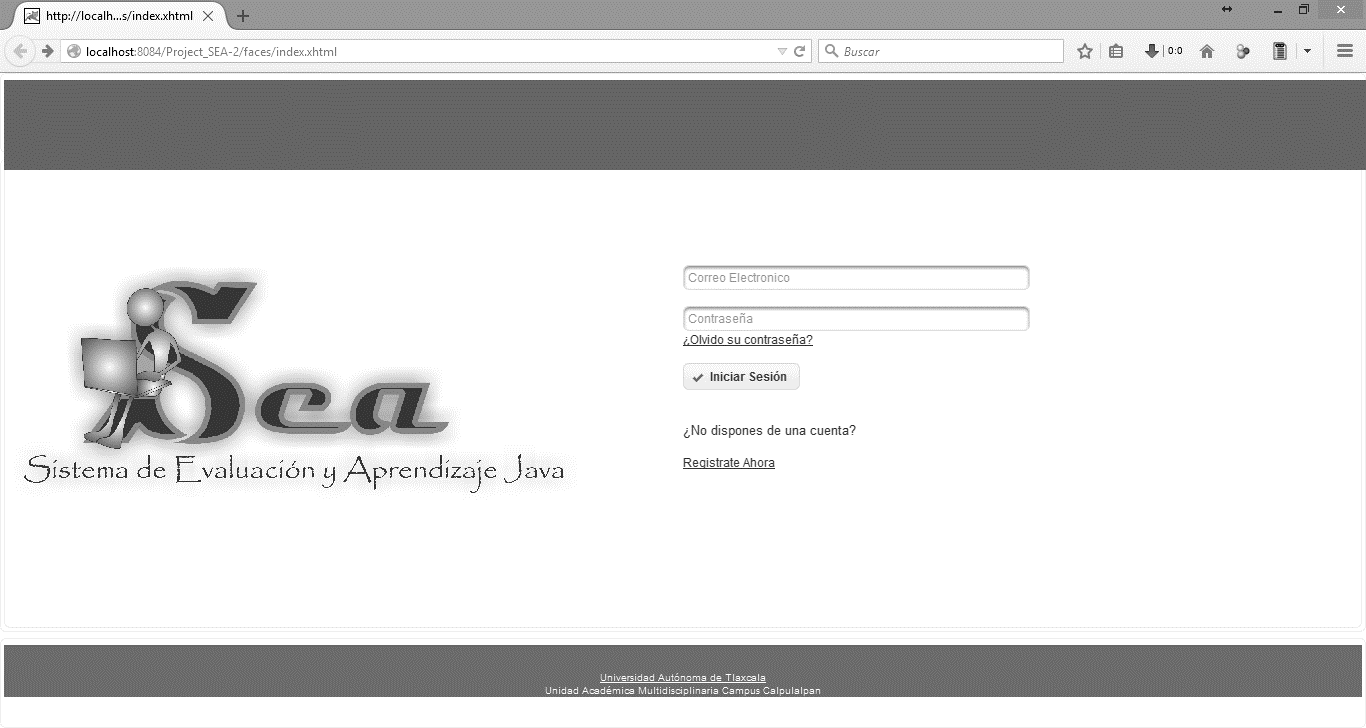
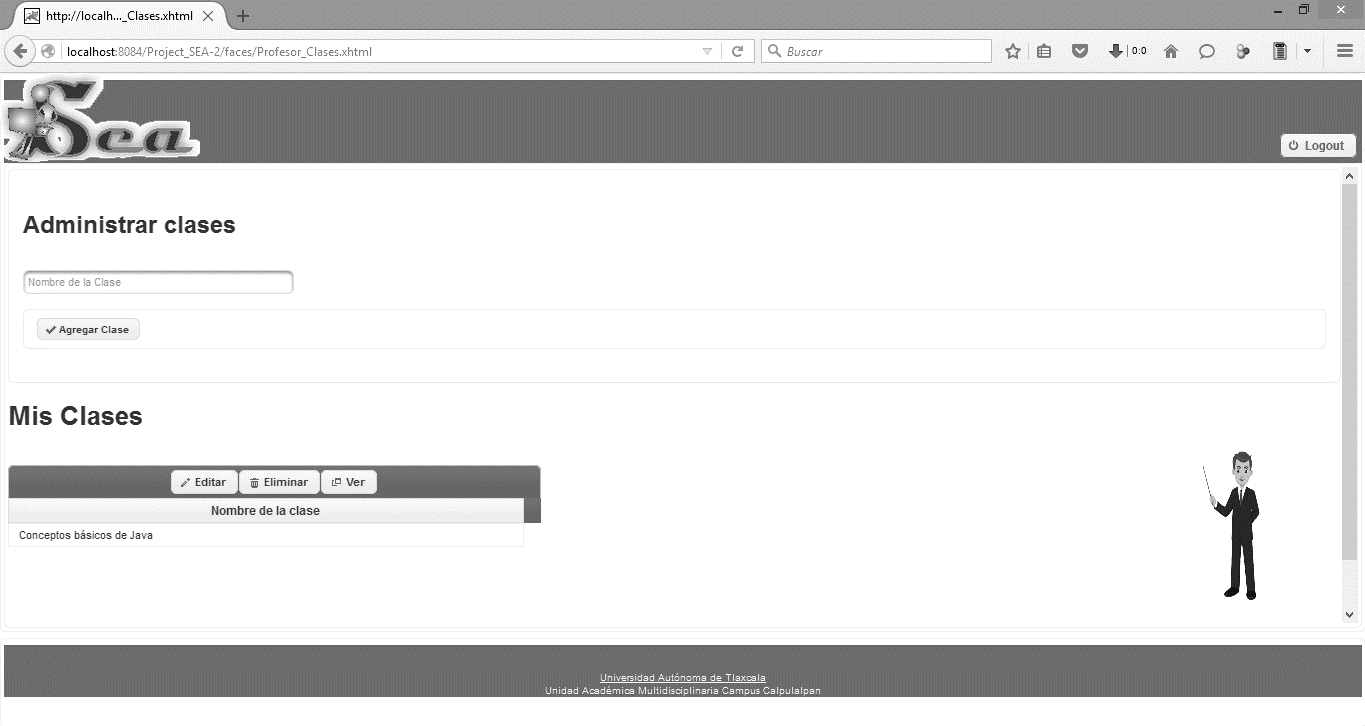


Figura 38 Interfaz de usuario SEA- CU-003-00 Administrar sesiones.

### SEA- CU-004-00 Administrar clases.

En la Figura 39 se muestra la interfaz de usuario del sistema para crear, editar, eliminar y visualizar las clases que creará el usuario (profesor).

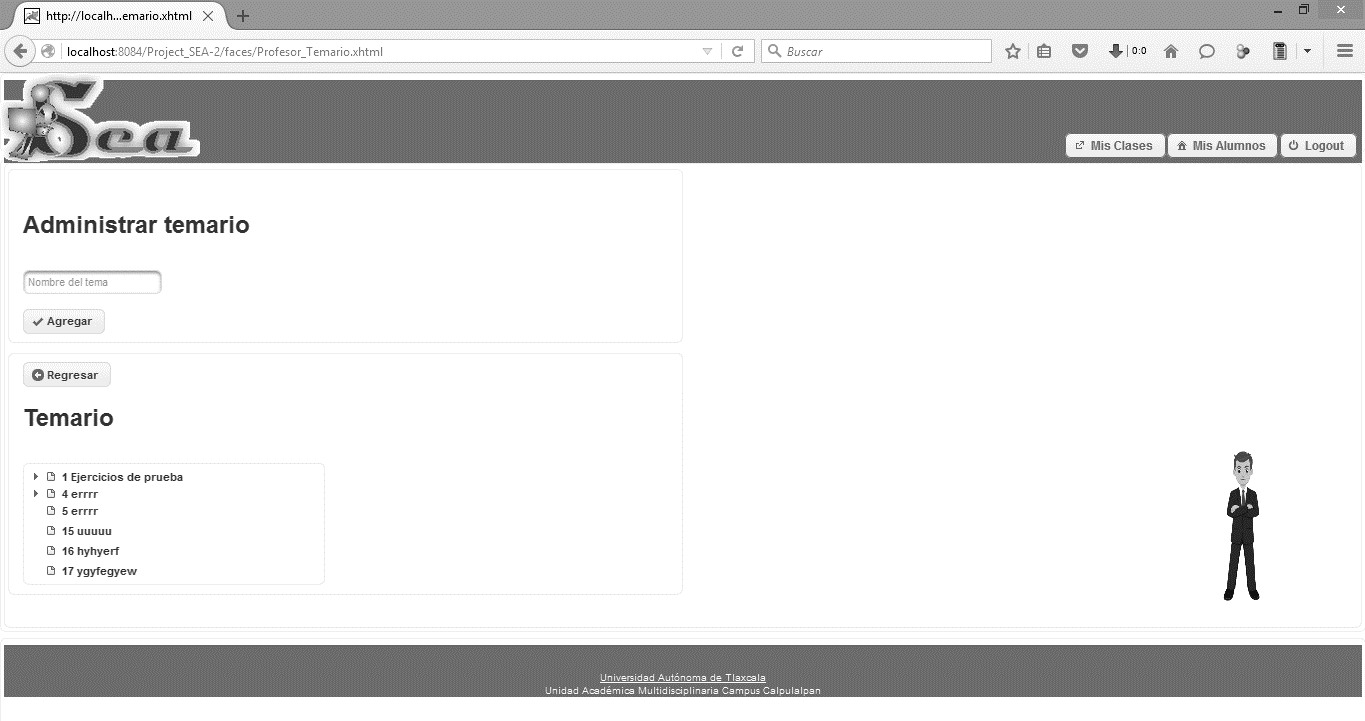
Figura 39 Interfaz de usuario SEA- CU-004-00 Administrar clases.



### SEA-CU-005-00 Administrar temario.

En la Figura 40 se muestra la interfaz de usuario del sistema para agregar, editar, eliminar y visualizar el temario.

Figura 40 Interfaz de usuario SEA-CU-005-00 Administrar temarios



### SEA- CU-006-00 Administrar ejercicios.

En la Figura 41 se muestra la interfaz de usuario del sistema para agregar un ejercicio al subtema seleccionado.

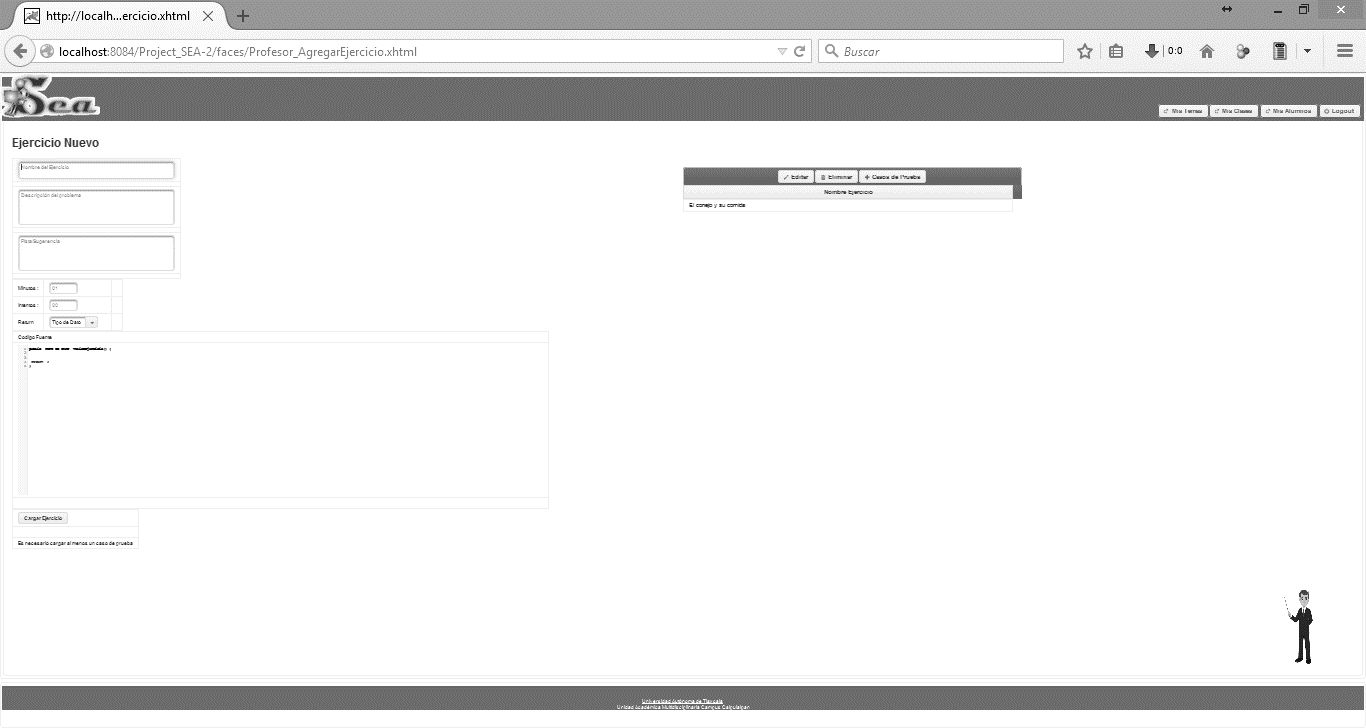


Figura 41 Interfaz de usuario SEA-CU-006-00 Administrar Ejercicio.

En la Figura 42 se muestra la interfaz de usuario de como agregar un caso de prueba a un ejercicio.

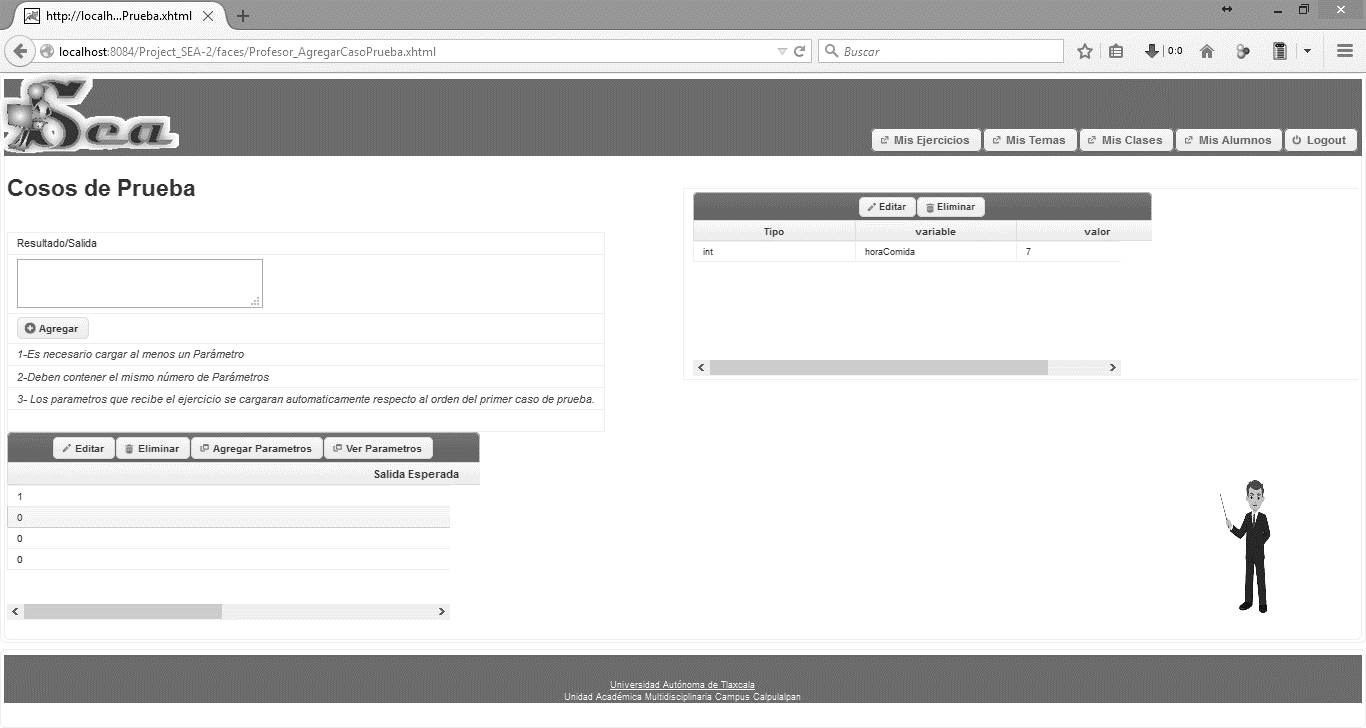


Figura 42 Casos de prueba SEA-CU-001-00 Administrar Ejercicio

### SEA- CU-007-00 Inscribirse a una clase.

En la Figura 43 se muestra la interfaz de usuario del sistema para que el alumno se inscriba a una clase que anteriormente el usuario “Profesor” ha creado en el sistema.

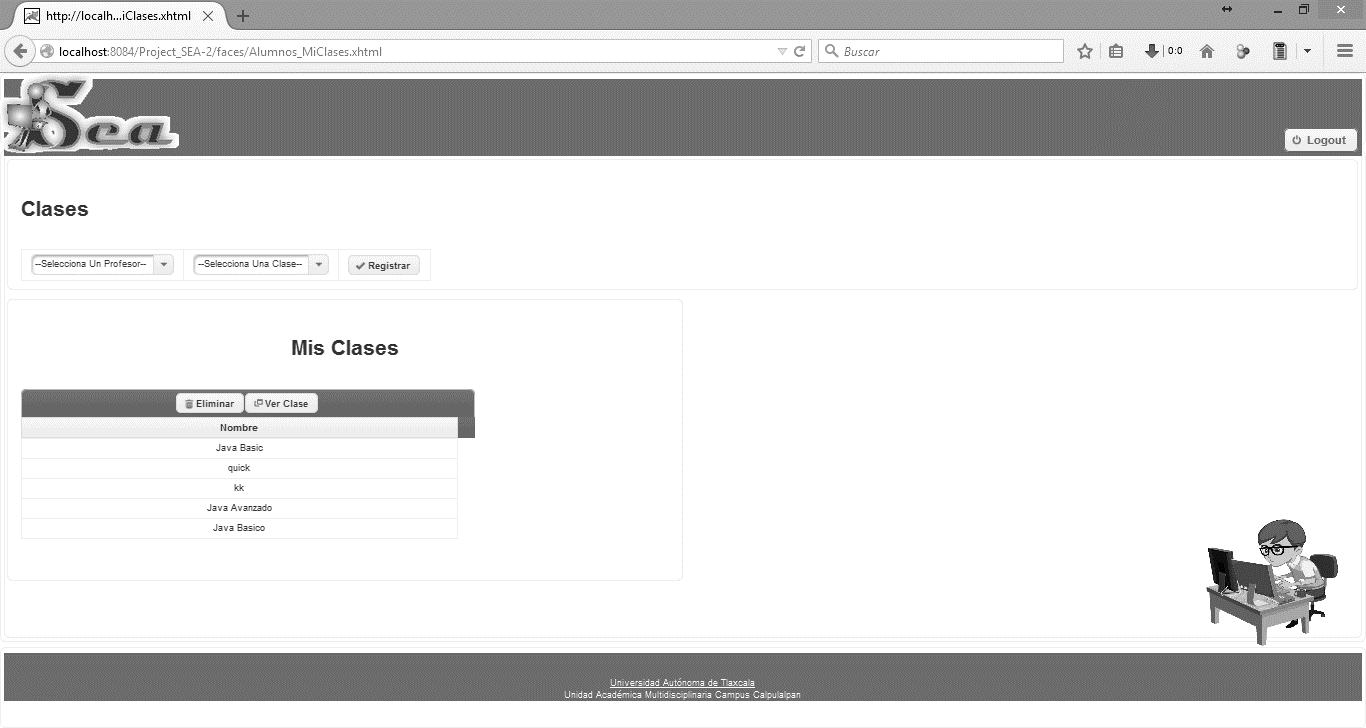


Figura 43 Interfaz de usuario SEA-CU-007-00 Inscribirse a una clase.

### SEA- CU-008-00 Resolver ejercicios

En la Figura 44 se muestra la interfaz de usuario del que listará el temario creado en una clase.



Figura 44 Interfaz de usuario SEA-CU-008-00 Resolver ejercicios.

En la Figura 45 se muestra la interfaz de usuario para listar los ejercicios que hay en un subtema.

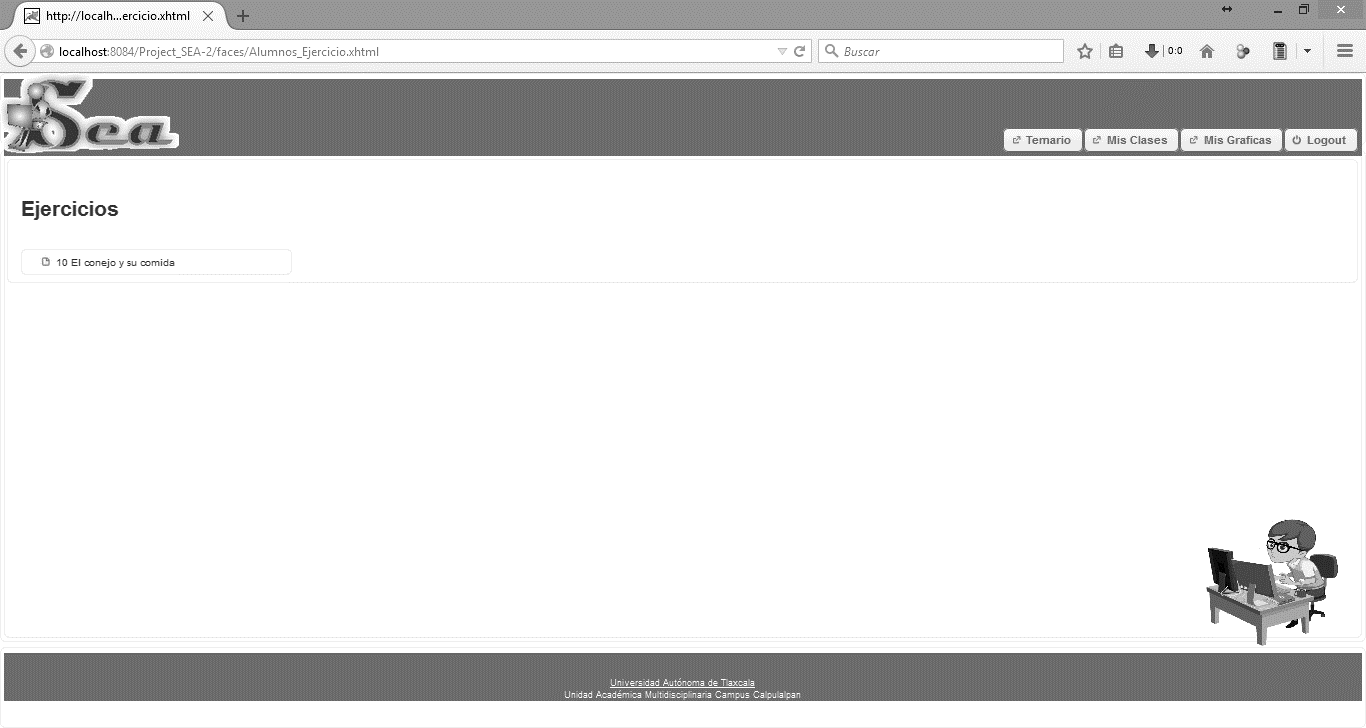


Figura 45 Interfaz de usuario SEA-CU-008-00 Resolver ejercicios (Ejercicios).

En la Figura 46 se muestra la interfaz de usuario dónde el usuario alumno debera dar solución a los ejercicios de codificación.

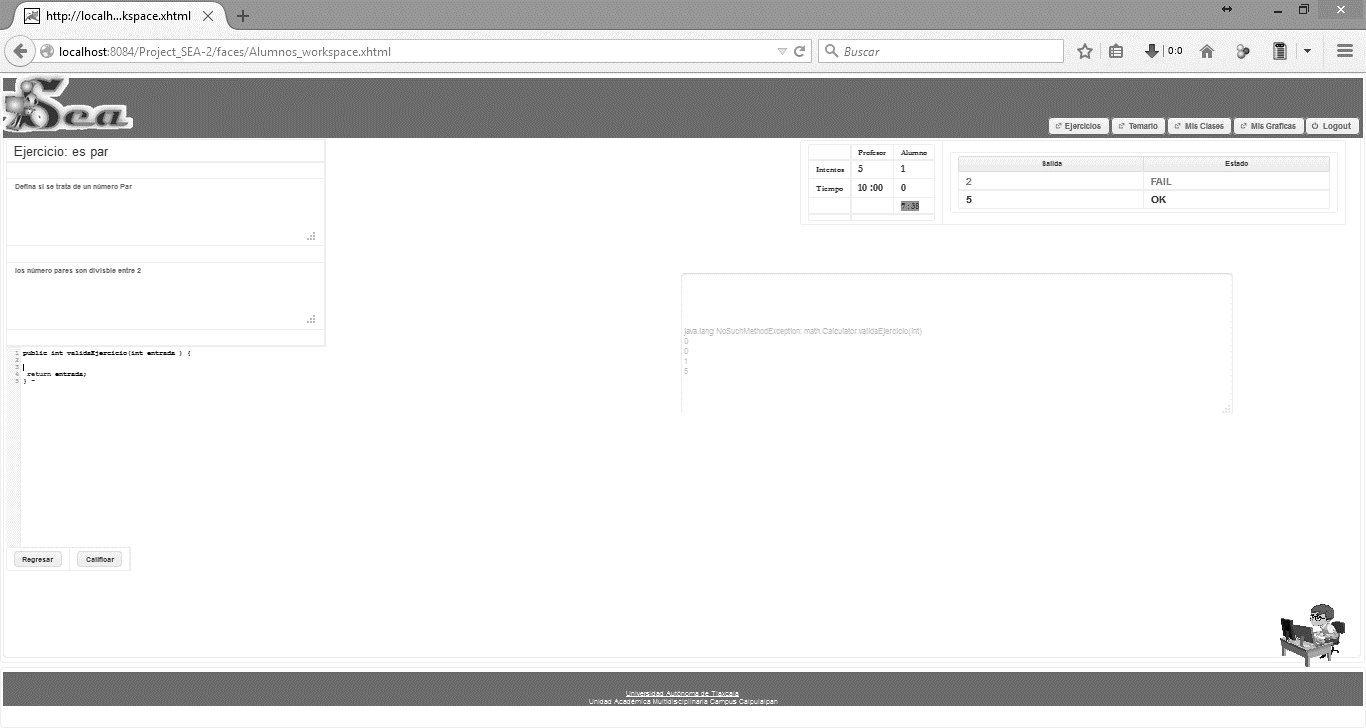


Figura 46 Interfaz de usuario SEA-CU-008-00 Resolver Ejercicio

### SEA- CU-009-00 Administrar avances

En la Figura 47 se muestra la interfaz de usuario del sistema para visualizar las gráficas de los intentos y tiempo que se han invertido en cada ejercicio.

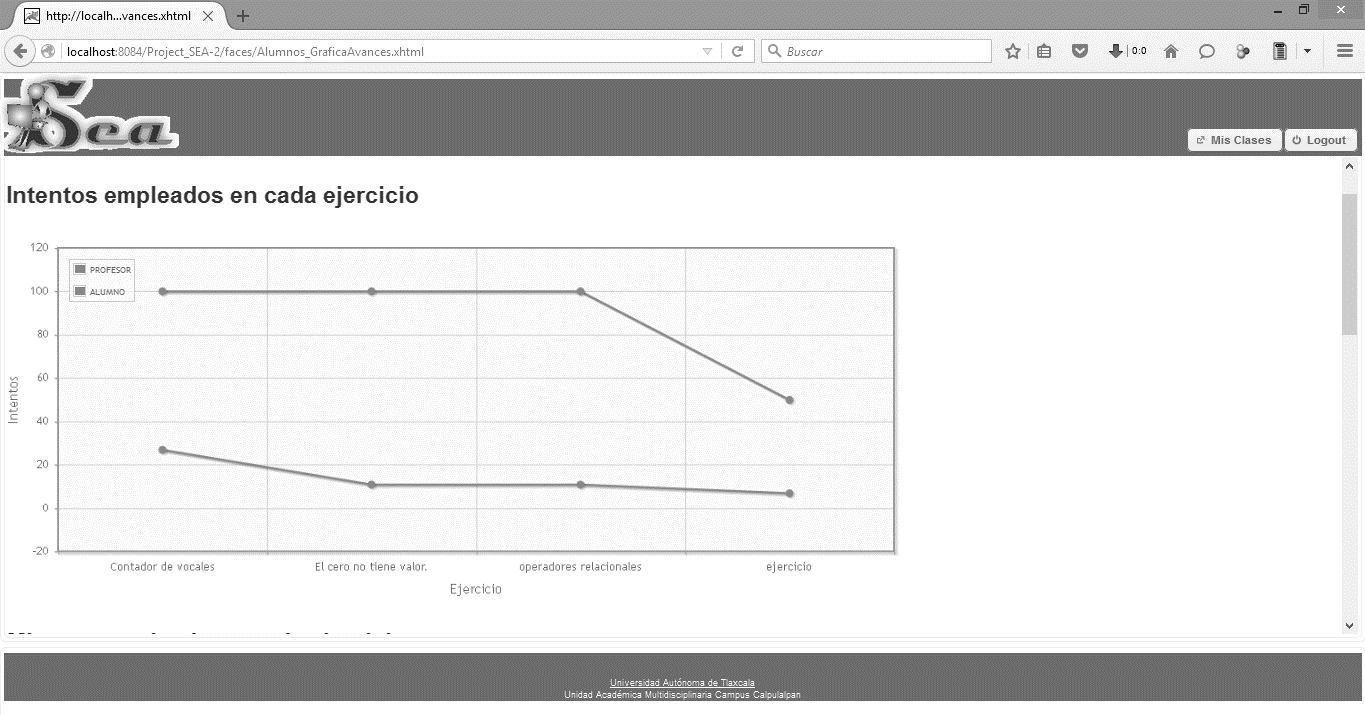


Figura 47 Interfaz de usuario SEA-CU-009-00 Administrar Avances.

En la Figura 48 se muestra la interfaz para visualizar los reportes en PDF de los avances que el alumno lleva en cada ejercicio, en las siguientes categorías: clase, tema, subtema y ejercicio.

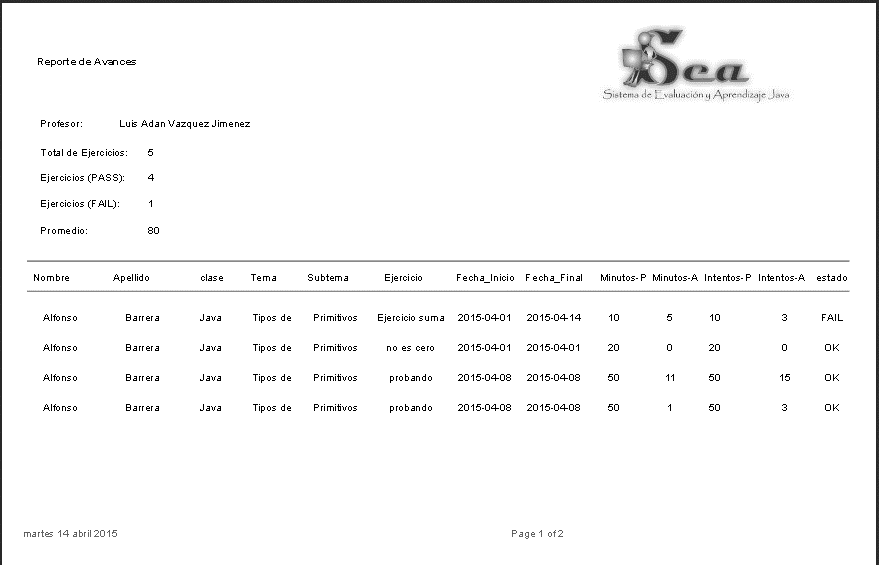


Figura 48 Interfaz de usuario SEA-CU-009-00 Administrar Avances (Reportes PDF)

La Figura 49 muestra la interfaz de usuario para visualizar las gráficas que llevan los avances de los ejercicios, los alumnos inscritos y los reportes de avances en PDF.

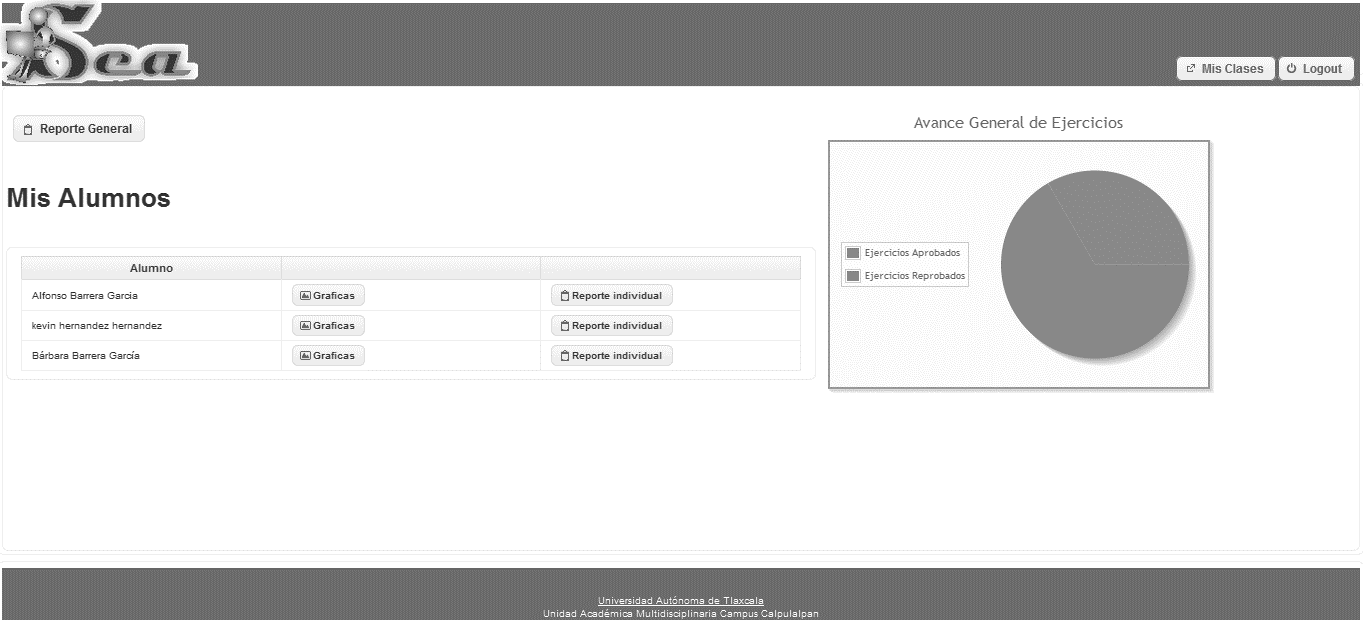


Figura 49 Interfaz de usuario SEA-CU-009-00 Administrar avances (Gráficas)

### SEA- CU-009-00 Servicios disponibles.

A continuación en la Tabla 18 se aprecian los servicios que ofrece el sistema SEA y la estructura que manejan, mostrando también una descripción de cada uno de ellos, siendo así más comprensible para el usuario final realizar una consulta utilizando el servicio deseado y disponible.

|  |  |  |
| --- | --- | --- |
| Servicio | Descripciones | Actor |
| Registrarse en SEA | Deberá proporcionar los datos correspondientes para crear una cuenta nueva en el sistema. | Profesor y alumno. |
| Crear clase | Se asignará el nombre a una clase para que un grupo de usuarios (Alumno) se inscriba. | Profesor. |
| Administrar temarios | Se crean temarios y subtemas para organizar los ejercicios. | Profesor. |
| Administrar Ejercicios | Se crea la estructura de un ejercicio, empezando con el nombre, descripción, tiempo e intentos límite y casos de prueba. | Profesor. |
| Gráficas de avance | Se muestra un conjunto de gráficas para visualizar el avance general del tiempo e intentos invertidos en cada ejercicio. | Alumno y Profesor. |
| Reportes de avance | Se generan reportes en PDF para visualizar a detalle el tiempo, intentos y si se aprobó en ejercicio en las siguientes categorías: clase, tema, subtema, ejercicio. | Alumno y Profesor. |

Tabla 18 Servicios Disponibles

### SEA- CU-009-00 Modelo de base de datos SEA

A continuación en esta sección se mostrará el diseño de la base de datos con el nombre de SEA que tendrá como finalidad el manejo de la información del sistema y la aplicación monitor. Para ello es necesario presentar el modelo E-R de la base de datos.

**3.7.1 Diagrama del modelo E-R- de SEA**

A continuación en la Figura 50 se muestra el diagrama E-R del sistema SEA.

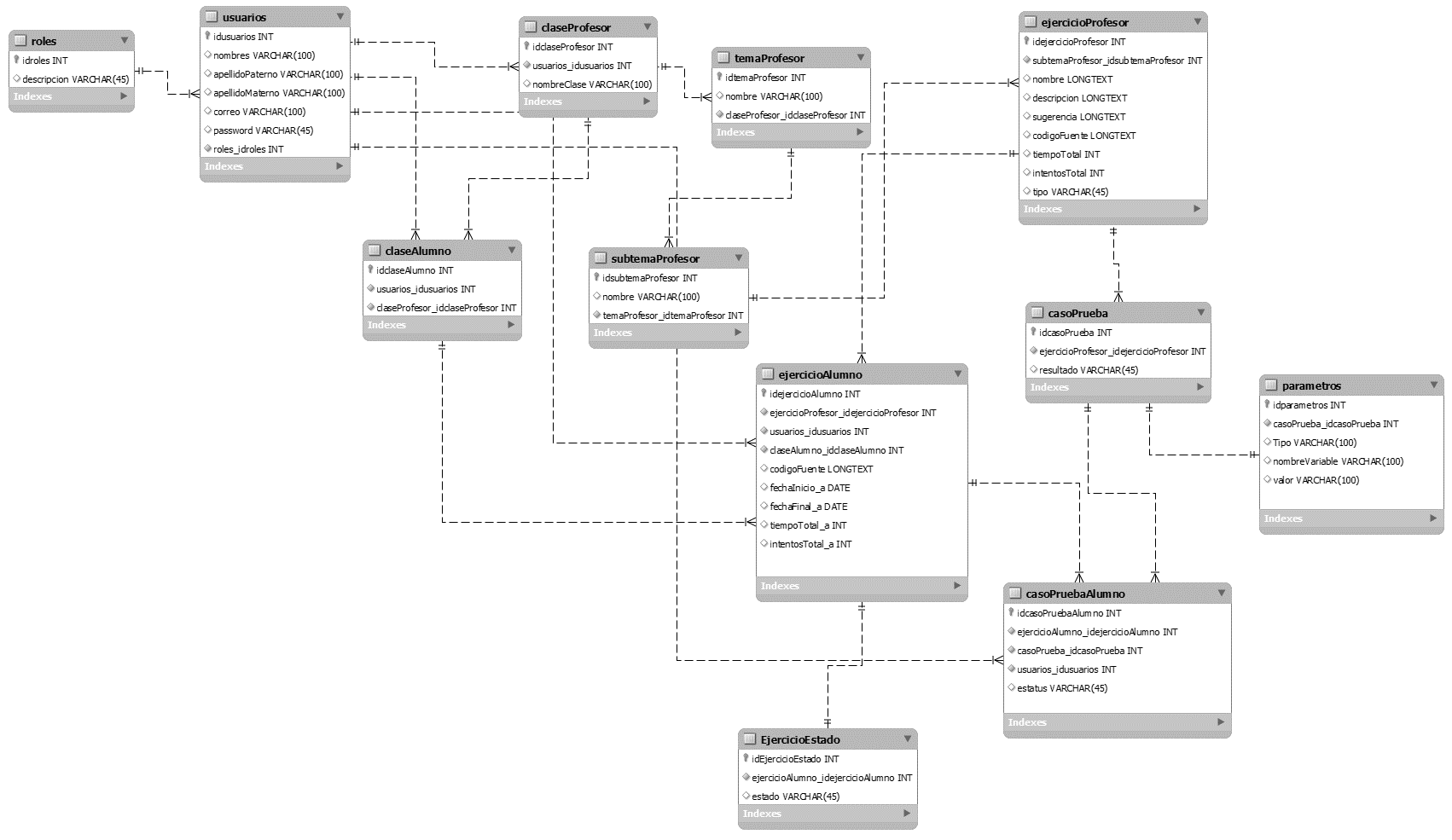


Figura 50 Diagrama del modelo E-R de SEA

# Implementación del Sistema

## Introducción

Dado el análisis que se ha realizado en el Capítulo 3 para el desarrollo de este sistema, a continuación se verá la implementación del sistema, que es el objetivo principal de esta investigación. Se mostrarán los procesos de desarrollo desde la creación de base de datos hasta la ejecución y despliegue del programa, siendo así un sistema para ser utilizado por usuarios finales a los que está enfocado.

Para mejor comprensión en el desarrollo de este sistema se montarán secciones de código fuente que tienen un funcionamiento importante dentro del sistema, permitiendo al lector tener una visión del ambiente del desarrollador.

## Configuración del ambiente de desarrollo.

En esta sección se proporciona información acerca del modo de configuración del entorno de desarrollo que permitirá codificar los elementos necesarios para el sistema.

### Java Platform JDK.

En primer lugar se debe de adquirir JDK en la página oficial que se muestra en la siguiente referencia [www.oracle.com/technetwork/es/java/javase/downloads/index.html](http://www.oracle.com/technetwork/es/java/javase/downloads/index.html), en descargas damos clic en Java Platform(JDK) la versión que esté vigente como se muestra en la



Figura 51 JDK

Se cargará una nueva ventana con las versiones que existen y en los sistemas operativos que es soportado. Aceptamos los términos de licencia para poder descargar la versión que soporte Windows 32 o 64 Bits como se muestra en la Figura 52.

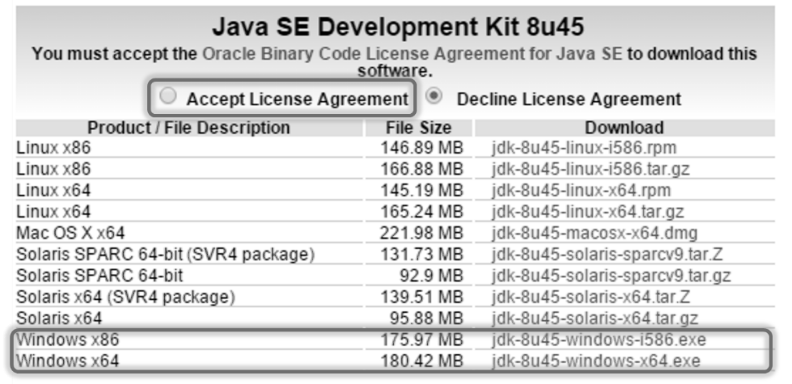


Figura 52 Descarga de JDK

Una vez que se ha descargado el archivo, procedemos con la instalación típica. Como se muestra en la Figura 53.

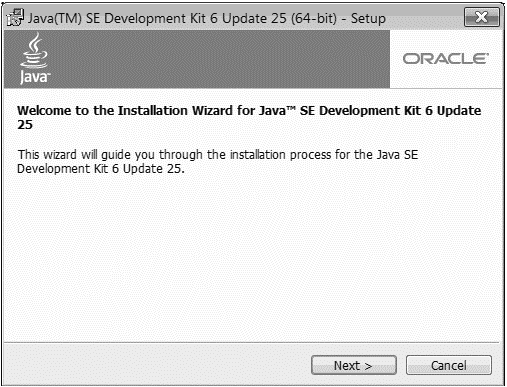


Figura 53 Instalación de JDK.

Continuamos con los detalles de la instalación como paquetes que se instalarán y la ubicación de los archivos Figura 54.

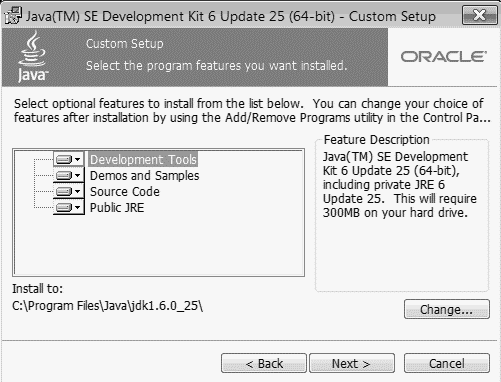


Figura 54 Configuración de la instalación.

Termina la instalación y se nos preguntará si deseamos instalar la máquina virtual de java JRE, presionamos Next. (Figura 55).

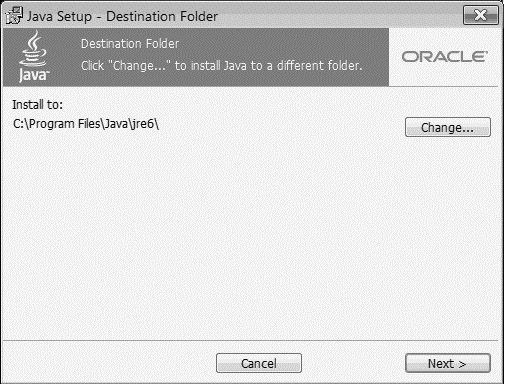


Figura 55 Instalación de máquina virtual de java.

Si se han seguido los pasos correctamente, se visualizará una ventana como la que se muestra en la Figura 56.

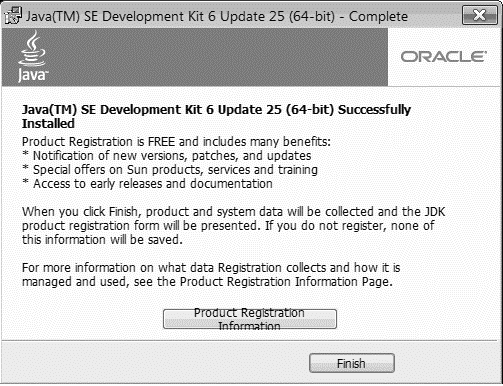


Figura 56 Instalación finalizada.

### IDE NetBeans.

En segundo lugar se debe adquirir el IDE NetBeans de la página oficial que se muestra en la siguiente referencia <https://netbeans.org/>, en Downloads, se adquiere la versión completa como se muestra en la .

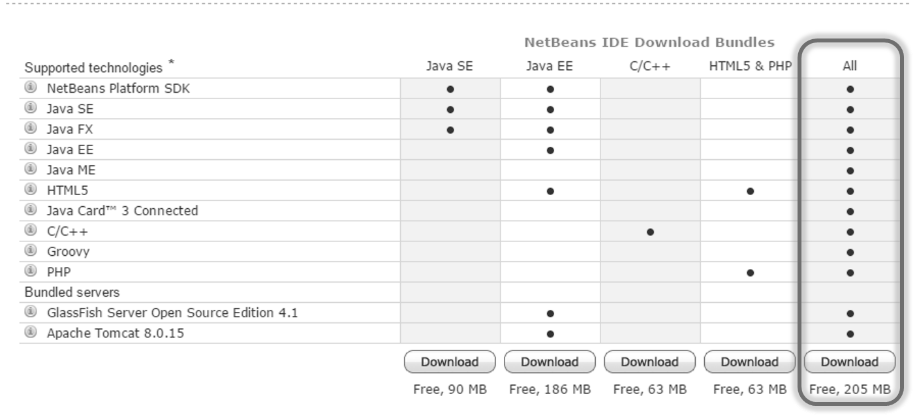


Figura 57 Descarga de NetBeans

Después se procederá con la instalación del IDE NetBeans dónde también incluiremos Apache Tomcat como se observa en la Figura 58.

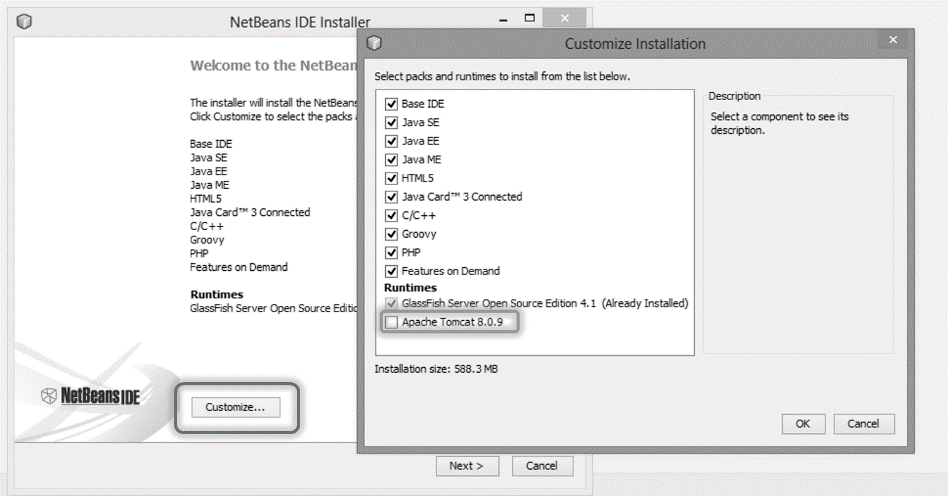


Figura 58 Incluir la instalación de Tomcat.

Una vez configurada la instalación proseguimos con la siguiente ventana (Figura 59).

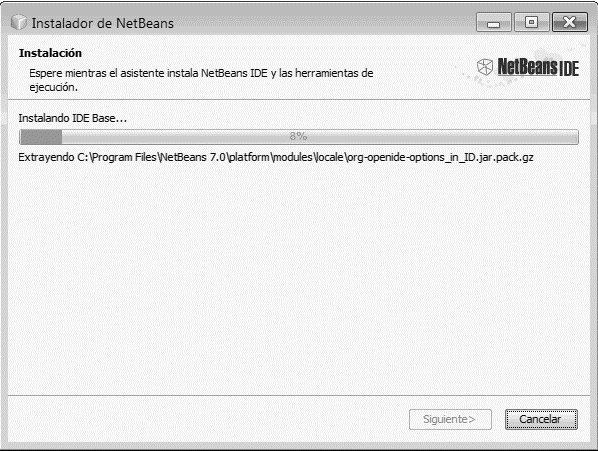


Figura 59 Instalación de NetBeans

El asistente de instalación nos preguntará si permitimos que se recopilen estadísticas sobre nuestro uso y finalizamos con la instalación de NetBeans. (Figura 60).



Figura 60 Instalación finalizada.

### Gestor de bases de datos MySQL

Se optó por utilizar MySQL 5.6 como sistema de gestión de bases de datos para el sistema SEA, el cual se descargará de la página oficial <https://dev.mysql.com/downloads/mysql/>, y después se procederá a su instalación ().

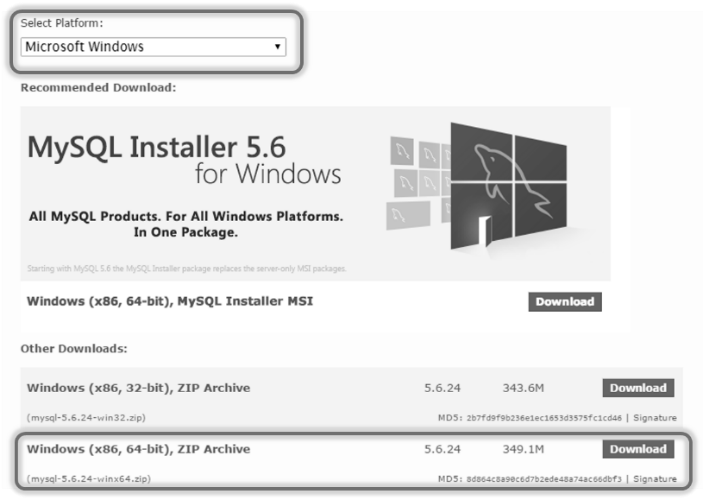


Figura 61 Descarga de MySQL Installer.

Iniciamos con la instalación de MySQL ver Figura 62.

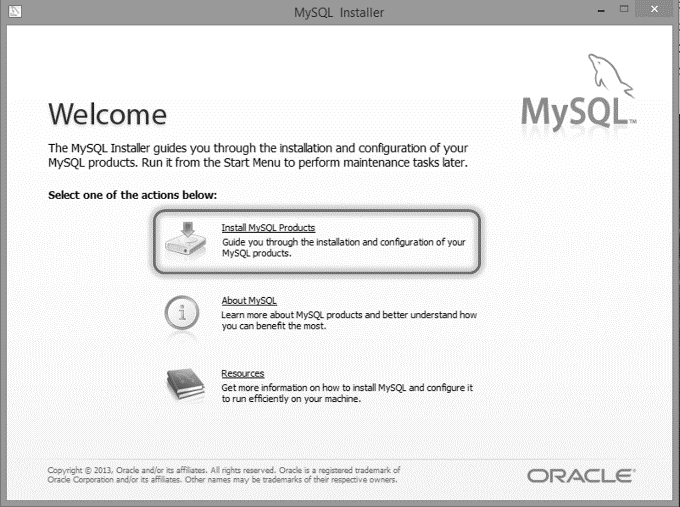


Figura 62 Instalación e MySQL.

Aceptamos los términos de la licencia para poder continuar con la instalación Figura 63

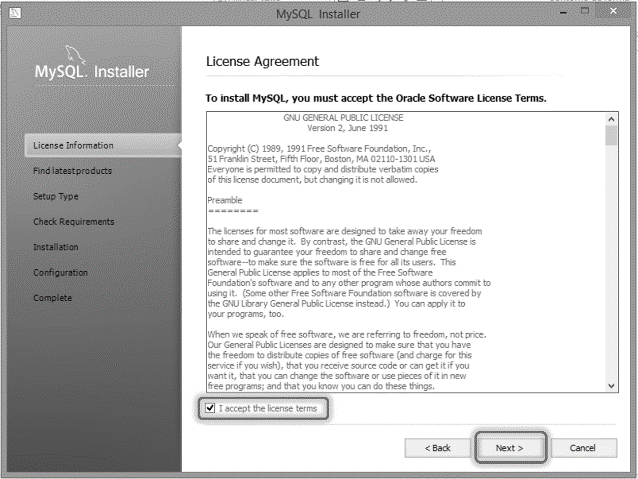


Figura 63 Términos de licencia.

Se selecciona el tipo de instalación en este caso se optó por la opción Full como se observa en la Figura 64.

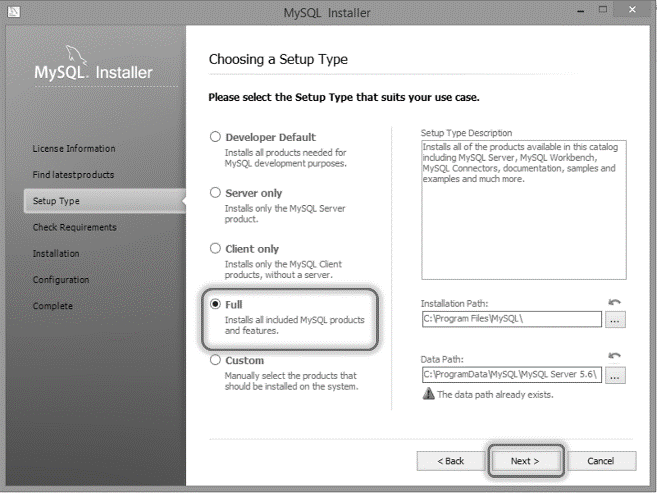


Figura 64 Tipos de instalación.

Se instalaron las actualizaciones y diferentes paquetes, pulsamos Next para continuar la instalación. Figura 65.

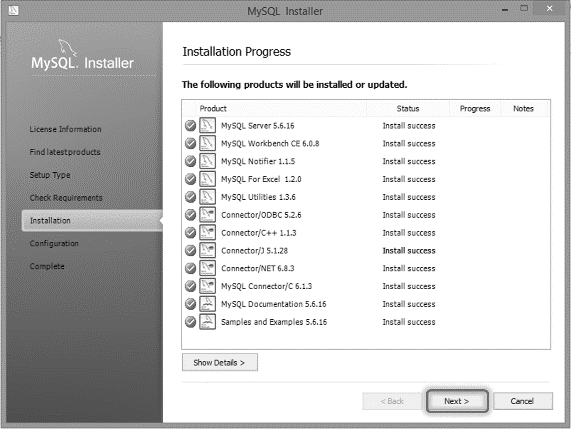


Figura 65 Progreso de instalación.

En la Figura 66 se observa las configuraciones, se dejaron las configuraciones que trae por defecto el asistente de instalación.

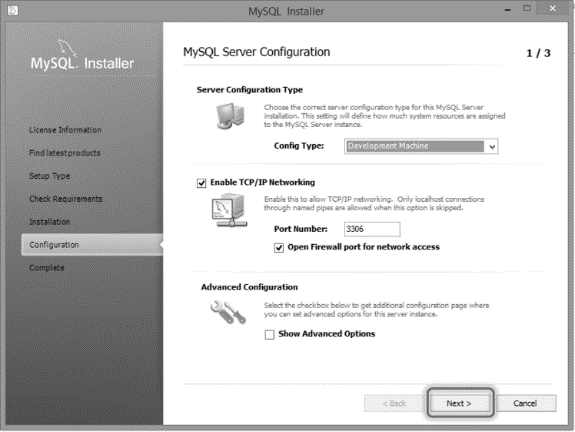


Figura 66 Configuración de instalación.

Como se observa en la Figura 67 se debe ingresar la contraseña para este caso el usuario root, En este caso crearemos de una vez el usuario para la creación de la base de datos SEA.

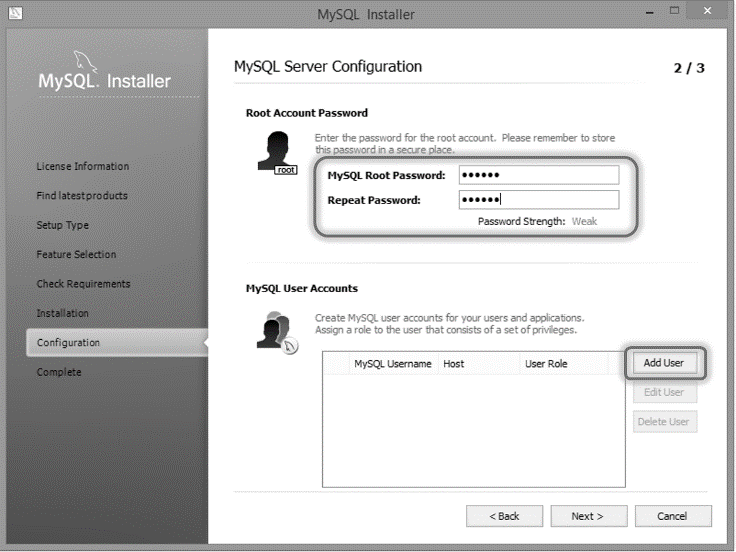


Figura 67 Asignación de contraseña para usuario root y crear un usuario.

Se escribe el nombre del usuario en este caso es “seadb”, se le asigna los permisos correspondientes y la contraseña para este usuario ver Figura 68.

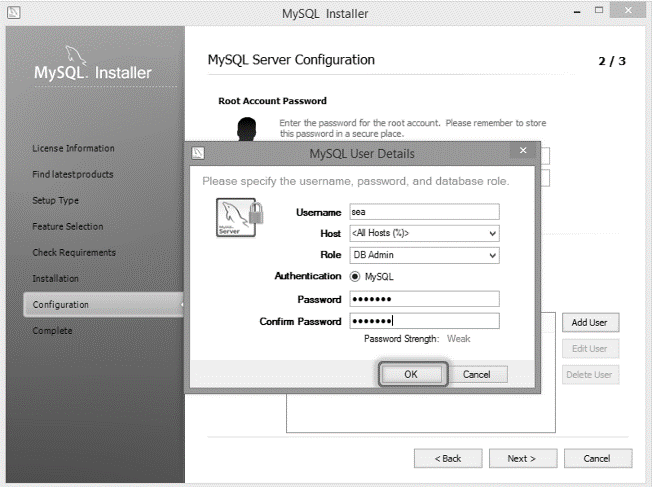


Figura 68 Asignación Permisos y contraseña al usuario “seadb”.

Se ha finalizado con la instalación y ahora ya se tiene un gestor de DB. Para la creación de la base de datos del SEA. (Figura 68).

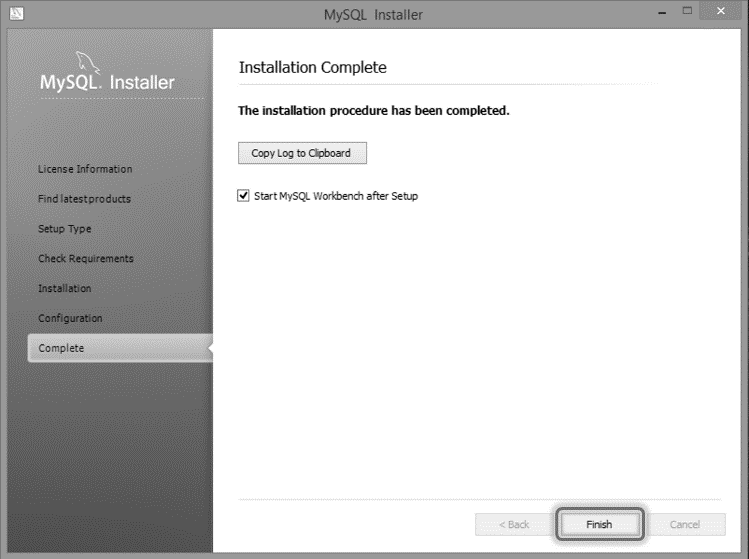


Figura 69 Finalización de la instalación.

### Creación de la base de datos.

Durante el proceso de instalación de MySQL se ha creado un usuario nombrado “seadb” como se puede apreciar en la figura 21. Ahora abrimos cmd, en Windows se accede al menú inicio, ejecutar y se escribe cmd y pulsamos enter. Una vez que se ha entrado localizamos la ruta de instalación de MySQL y dentro de la carpeta bin ejecutamos el comando: mysql –u seadb –p como se puede observar en la Figura 70, y por ultimo escribimos la contraseña.

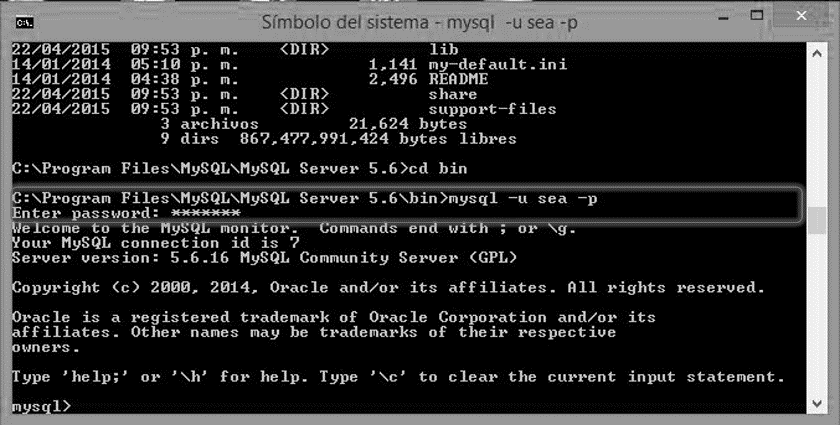


Figura 70 Entrando a la Base de datos con el usuario “seadb”.

Finalmente solo queda ingresar el script generado desde la aplicación Workbench dónde se tiene el modelo E-R del que se aprecia en la figura 3.41,del apartado anterior. El script puede ser consultado en el apéndice: A.1 Script de Base de datos para el sistema SEA y la aplicación monitor.

### Muestreo de tablas en MySQL en ventana de comandos.

En la se muestra las tablas creadas desde la DBMS **Workbench**, aplicando ingeniería inversa según el diagrama E-R de la figura 3.41 del apartado anterior, dando un total de 12 tablas según los requerimientos del sistema. El script puede ser consultado en el Apéndice: A.1 Script de base de datos del sistema SEA.

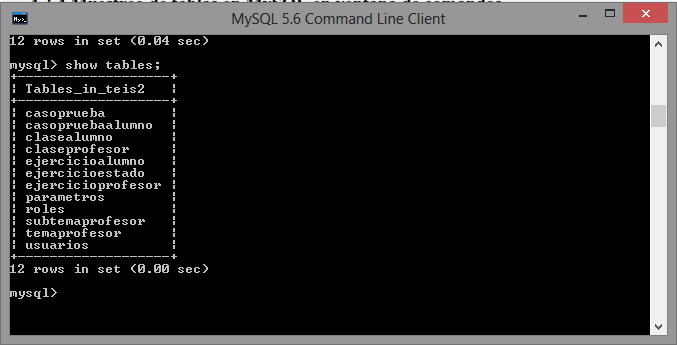


Figura 71 Tablas de la base de datos SEA.

## Para el desarrollo del sistema SEA.

Para el desarrollo del sistema se utilizaron los códigos que se citan en el apéndice B, de los cuales se comprende la lógica de negocios, generación de vistas para el usuario final y acciones particulares como envíos de mensajes electrónicos y la compilación de código de los ejercicios. A continuación se harán referencia a las clases según su implementación para cada caso de uso, basados en los diagramas de clases, los diagramas de secuencia y los diagramas de estado.

Para la implementación del conjunto de casos de uso del sistema SEA se desarrollaron los siguientes componentes:

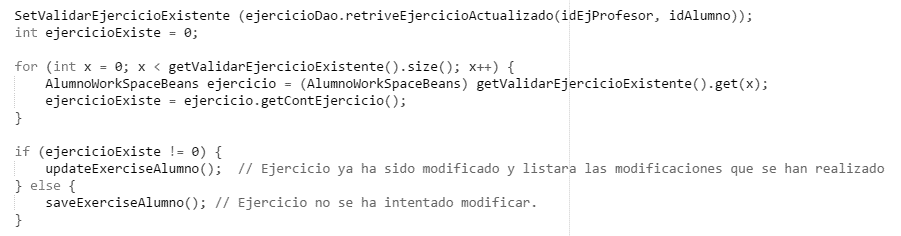
1. Los Beans son usados principalmente para encapsular los datos de negocio. Se utiliza una única llamada a un método para enviar y recuperar los datos. Con el fin de encapsulación de datos de negocio de crearon los beans necesarios para tener una mejor transferencia de información los cuales se citan en el apéndice B.1 tabla 6.2.
2. Los Dao para implementar las operaciones básicas sobre las entidades que componen la aplicación. La funcionalidad ofrecida por este conjunto corresponde de un modo genérico con el patrón DAO (Data Access Object), que oculta las tecnologías y el modo de acceso a los datos, delegando, en este caso las operaciones concretas. Se hacen las 4 operaciones básicas (crear, buscar por ID, actualizar y borrar). Se citan en el apéndice B.1 tabla 6.3.
3. La clase MysqlConnect se utiliza para manejar la conexión a la base de datos y mantener o cerrar las sesiones que se utilizarán en el DAO.
4. Los ManagedBean implementan los casos de uso específicos de la aplicación. En general proveen de operaciones de mayor complejidad que las de paquete DAO, responsabilizándose de coordinar las invocaciones de otros EJBs encargados del manejo de datos. Todos ellos tienen alcance de sesión (*@SessionScoped*), por lo que sus atributos estarán disponibles mientras dure la sesión del usuario. Se citan en el apéndice B.1 tabla 6.4.
5. Se crearon las vistas con la finalidad de tener una interfaz que le permita a los usuarios poder interactuar con el sistema SEA. Las vistas creadas se citan en el apéndice B.1 tabla 6.5. la capa de presentación web se ha implementado el Framework JSF con los ficheros XHTML dónde de define la disposición y propiedades de los componentes JSF de la presentación web.

## Desarrollo de la aplicación.

A continuación en esta sección se verá a grandes rasgos la implementación de una aplicación que permitirá compilar y ejecutar los ejercicios resueltos por el alumno y compararlo con los casos de prueba que el profesor ha establecido, también verificará que no supero el número de intentos y tiempo asignado. La lógica que el alumno implemente y califique será almacenada en la base de datos junto con el tiempo e intentos que ha invertido en resolver el ejercicio.

### Compilación y ejecución del código.

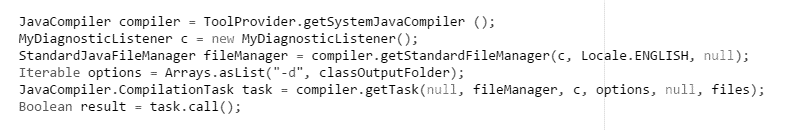
Una vez que el alumno ha decidido resolver un ejercicio, el sistema cargará los datos almacenados de la BD, comprobará si el ejercicio ya se ha intentado resolver y listará las modificaciones que ha hecho el alumno, en caso que el ejercicio sea la primera vez que se abre, carga los datos que existen en la base de datos del profesor.



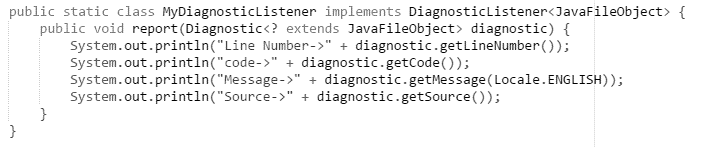
Una vez que se han cargado los datos el alumno procederá a resolver el ejercicio y cuando crea que la lógica que ha implementado es capaz de superar los casos de prueba se calificará. Como primer paso se creará un archivo en memoria que contenga la lógica, como se muestra en el código a continuación.



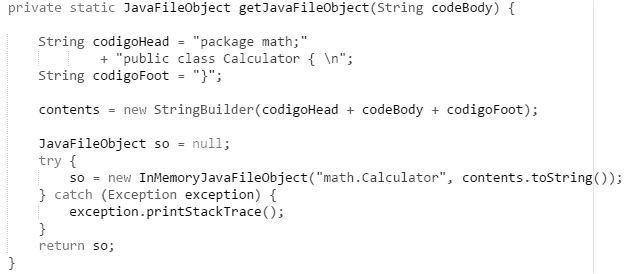
Después obtendremos el compilador de java (JavaCompiler), y el recolector de problemas del sistema (MyDiagnosticListener), luego usaremos StandardJavaFileManager para personalizar como el compilador leerá y escribirá el archivo, especificamos la carpeta de salida para la clase, como se muestra en el siguiente código.



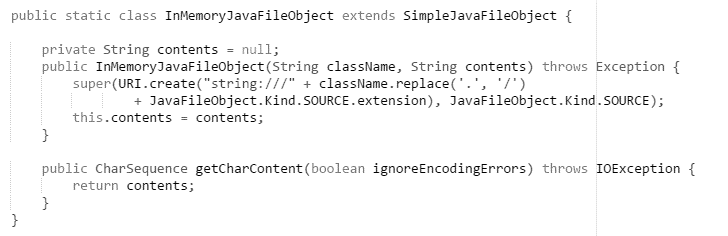
En el siguiente código MyDiagnosticlistener recolectará los errores sintácticos o semánticos que se generen a la hora de compilar el código; En caso de existir un error mostrará en que línea ocurrió, el código que provoco el error y el mensaje del compilador.



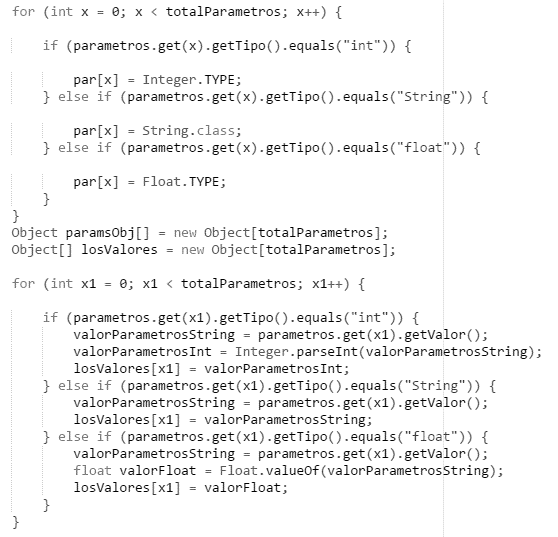
Se crea un StringBuilder para almacenar el código fuente del archivo, y se pasa ese parámetro al método InMemoryJavaFileObject.



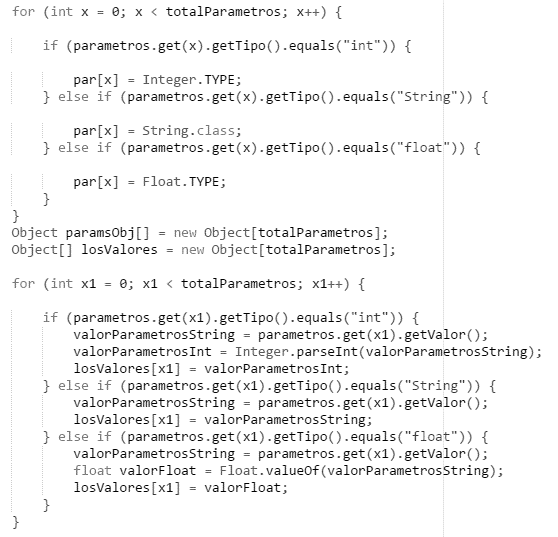
Java File Object representa un archivo de código fuente de java en la memoria, así que no hay necesidad de poner el archivo de código fuente en el disco duro. Y se convertirá en un URI, ya que es el formato que es esperado por JavaFileObject.



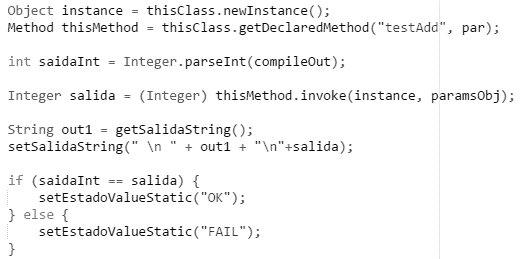
Se creará un arreglo del tipo Class para almacenar el tipo de dato (int, String o float).



Se asigna el valor al arreglo dependiendo si es int, String o float, con los valores de la base de datos. Los valores son listados de los parámetros que el profesor asignó para este caso de prueba.



Es ejecutado con los valores y se captura la salida del compilador, se comparará con la que se espera que sea la respuesta, declarada anteriormente por el profesor.



# Resultados

## Introducción

En este capítulo se mostrarán los resultados obtenidos durante las pruebas realizadas una vez finalizada la implementación, para ello los resultados se mostrarán en pequeños casos de prueba para cada función existente en el sistema, describiendo así la función que se realizó y el resultado que se obtiene.

Además, podremos tener una apreciación estadística del funcionamiento general del sistema, es decir, podremos saber si puede ser apto para ser implementado en un futuro y proporcionar las ventajas cumpliendo los objetivos para los que fue desarrollado.

## Casos de prueba.

Los casos de prueba son importantes para el sistema ya que en estos podremos visualizar de forma resumida los resultados de cada funcionalidad que existe en este sistema, indicando así el porcentaje aproximado de fiabilidad real al que está sujeto cada funcionalidad.

### SEA-CP-001-00 Registrar Alumno/Profesor

En Se muestra los resultados esperados de las pruebas de acuerdo a SEA-001-00 crear cuenta alumno/profesor.

|  |  |
| --- | --- |
| Propósito | Registrar alumno/profesor al sistema. |
| Pre-requisitos | No estar registrado con la misma cuenta de correo electrónico. |
| Acciones | 1.-Seleccionar la opción “Regístrate Ahora” que se encuentra en la pág. Principal.  2.-Ingresar la información solicitada.  3.-Escoger el perfil de alumno.  4.-Dar clic en el botón de “Guardar”. |
| Notas | Si las acciones son efectuadas correctamente la funcionalidad del registro en el sistema será 100% exitosa. |
| Resultados esperados | Registro del alumno exitoso. |
| Resultados reales. | Exitoso 100%. |
| Fecha | 17 de febrero de 2015, 08:25:13 p.m. |

Tabla 19 SEA-CP-001-00 Registrar Alumno/Profesor

### SEA-CP-002-00 Recuperar contraseña

En la se muestran los resultados esperados de acuerdo a SEA-002 Recuperar Contraseña.

|  |  |
| --- | --- |
| Propósito | Recuperar la contraseña para acceder al sistema. |
| Pre-requisitos | Estar registrado en el sistema. |
| Acciones | 1.-Seleccionar la opción “Olvido su contraseña” que se encuentra en la pág. Principal.  2.-Ingresar correo electrónico.  3.-Escoger el perfil de alumno.  4.-Dar clic en el botón de “Buscar”. |
| Notas | Si las acciones son efectuadas correctamente la funcionalidad de recuperar contraseña en el sistema será 100% exitosa. |
| Resultados esperados | Recuperación de la contraseña. |
| Resultados reales. | Exitoso 100%. |
| Fecha | 17 de febrero de 2015, 08:30:13 p.m. |

Tabla 20 SEA-CP-002-00 Recuperar contraseña.

### SEA-CP-003-00 Inicio de sesión.

En la se muestran los resultados esperados de acuerdo a SEA-003 Inicio de Sesión.

|  |  |
| --- | --- |
| Propósito | Iniciar sesión al sistema de SEA. |
| Pre-requisitos | Estar registrado en el sistema. |
| Acciones | 1.-Ingresar correo electrónica.  2.-Ingresar contraseña.  3.-Dar clic en el botón de “Iniciar sesión”. |
| Notas | Si las acciones son efectuadas correctamente la funcionalidad de iniciar sesión en el sistema será 100% exitosa. |
| Resultados esperados | Ingreso al sistema. |
| Resultados reales | Exitoso 100%. |
| Fecha | 17 de febrero de 2015, 08:40:00 p.m. |

Tabla 21 SEA-CP-003-00 Iniciar sesión.

### SEA-CP-004-00 Administrar clases

En la se muestran los resultados esperados de acuerdo a SEA-004 Administrar clases.

|  |  |
| --- | --- |
| Propósito | Administrar las clases del sistema SEA. |
| Pre-requisitos | Haber iniciado sesión en el sistema. |
| Acciones | 1.-Ingresar el nombre deseado a una clase.  2.-Esa clase puede ser eliminada o editada.  3.-La clase permitirá la creación de un temario. |
| Notas | Si las acciones son efectuadas correctamente la funcionalidad de Administrar clases en el sistema será 100% exitosa. |
| Resultados esperados | Creación, edición y eliminación de clases. |
| Resultados reales | Exitoso 100%. |
| Fecha | 17 de febrero de 2015, 09:00:00 p.m. |

Tabla 22 SEA-CP-004-00 Administrar clases.

### SEA-CP-005-00 Administrar temarios.

En la se muestran los resultados esperados de acuerdo a SEA-005 Administrar Temarios.

|  |  |
| --- | --- |
| Propósito | Administrar los temarios de las clases en el sistema SEA. |
| Pre-requisitos | Haber creado una clase. |
| Acciones | 1.-Ingresar el nombre deseado del tema y clic en el botón agregar.  2.-Ese tema puede ser eliminado o editado y a su vez se le puede agregar subtemas.  3.-Ingresar el nombre al nuevo subtema y clic en agregar.  4.-Los subtemas pueden ser editados, eliminados y poder agregar ejercicios. |
| Notas | Si las acciones son efectuadas correctamente la funcionalidad de Administrar temarios en el sistema será 100% exitosa. |
| Resultados esperados | Creación, edición y eliminación de temas y subtemas. |
| Resultados reales | Exitoso 100%. |
| Fecha | 17 de febrero de 2015, 09:20:09p.m. |

Tabla 23 SEA-CP-005-00 Administrar temarios.

### SEA-CP-006-00 Administrar ejercicios.

En la se muestran los resultados esperados de acuerdo a SEA-006 Administrar Ejercicios.

|  |  |
| --- | --- |
| Propósito | Administrar los ejercicios en el sistema SEA. |
| Pre-requisitos | Haber creado un temario. |
| Acciones | 1.-Ingresar el nombre, descripción, tiempo e intentos y clic en el botón cargar ejercicio.  2.-Los ejercicios pueden ser eliminados, editados y se podrá asignar casos de prueba.  3.- En los casos de prueba se asignará primero una salida o return que será evaluado.  4.-Cada salida deberá tener al menos un parámetro.  5.- El número de parámetros debe ser el mismo para todas las salidas del mismo ejercicio.  6.-Las salidas y parámetros pueden ser editados o eliminados. |
| Notas | Si las acciones son efectuadas correctamente la funcionalidad de administrar ejercicios en el sistema será 100% exitosa. |
| Resultados esperados | Creación, edición y eliminación de temas y subtemas. |
| Resultados reales | Exitoso 100%. |
| Fecha | 17 de febrero de 2015, 09:40:02p.m. |

Tabla 24 SEA-CP-006-00 Administrar Ejercicios.

### SEA-CP-007-00 Inscribirse a una clase.

En la se muestran los resultados esperados de acuerdo a SEA-007 inscribirse a una clase.

|  |  |
| --- | --- |
| Propósito | Inscribirse a una clase en el sistema SEA. |
| Pre-requisitos | 1.- Haber creado una cuenta como alumno en el sistema SEA.  2.- Que el usuario profesor haya creado una clase. |
| Acciones | 1.-Seleccionar un profesor.  2.-Seleccionar una clase del profesor.  3.-Clic en el botón registrar.  4.- Si el usuario alumno lo desea a la clase que se ha registrado, puede eliminarla. |
| Notas | Si las acciones son efectuadas correctamente la funcionalidad de inscribirse a una clase en el sistema será 100% exitosa. |
| Resultados esperados | Inscripción a una clase. |
| Resultados reales | Exitoso 100%. |
| Fecha | 17 de febrero de 2015, 09:40:01p.m. |

Tabla 25 SEA-CP-007-00 Inscribirse a una clase.

### SEA-CP-008-00 Resolver ejercicios.

En la se muestran los resultados esperados de acuerdo a SEA-008 Resolver ejercicios.

|  |  |
| --- | --- |
| Propósito | Resolver los ejercicios existentes en una clase. |
| Pre-requisitos | Haberse registrado a una clase. |
| Acciones | 1.-Aparecerá el área de trabajo para el alumno con las siguientes partes: nombre, descripción, pistas, intentos y tiempo límite asignados por el profesor para ese ejercicio, una tabla con los casos de prueba que se evaluarán, un cronometro.  2.-El alumno deberá implementar la lógica correspondiente para dar solución al ejercicio y dará clic al botón calificar para evaluar con los casos de prueba.  3.-Si el tiempo o el número de intentos son superados el botón de calificar se bloqueará. |
| Notas | Si las acciones son efectuadas correctamente la funcionalidad de resolver ejercicios en el sistema será 100% exitosa. |
| Resultados esperados | La solución exitosa de cada uno de los ejercicios sin superar el tiempo e intentos límite. |
| Resultados reales | Exitoso 100%. |
| Fecha | 17 de febrero de 2015, 09:50:19p.m. |

Tabla 26 SEA-CP-008-00 Resolver ejercicios.

### SEA-CP-009-00 Administrar avances.

En la se muestran los resultados esperados de acuerdo a SEA-009 Administrar avances.

|  |  |
| --- | --- |
| Propósito | Administrar Avances de las clases en el sistema SEA. |
| Pre-requisitos | Haber dado solución a los ejercicios. |
| Acciones | (Alumnos)  1.-Ingresar al botón mis gráficas.  2.-Se cargará automáticamente gráficas en las siguientes categorías: días, minutos e intentos invertidos en cada ejercicio.  3.-Clic en el botón ver reportes que mostrar un PDF con los detalles del ejercicio.  (Profesor)  1.-Ingresar al botón mis alumnos.  2.-Listará los alumnos inscritos a esa clase con la opción de ver sus reportes por clase, tema o subtema.  3.-Clic en el botón reporte general para visualizar reportes en PDF de todos los alumnos que se encuentren en esa clase, tema o subtema. |
| Notas | Si las acciones son efectuadas correctamente la funcionalidad de administrar avances en el sistema será 100% exitosa. |
| Resultados esperados | Visualizar gráficas y reportes. |
| Resultados reales | Exitoso 100%. |
| Fecha | 17 de febrero de 2015, 09:20:09p.m. |

Tabla 27 SEA-CP-009-00 Administrar avances.

# Conclusiones y trabajo futuro.

Finalmente se expresarán las conclusiones correspondientes a los capítulos anteriores que muestran los conceptos fundamentales que a su vez describen de manera general el ambiente distribuido, el desarrollo y la implementación del sistema.

## Conclusiones.

Para cumplir con los objetivos del sistema, la aplicación se dividió en dos partes, una para usuarios con perfil de profesor y la otra para los usuarios con perfil de alumno, dónde en la primera de ellas, el profesor después de haber creado una cuenta en el sistema deberá crear una clase, con sus respectivos temas y subtemas, ejercicios con sus cosos de prueba. Todo ello que cubra con los objetivos y necesidades de los alumnos a los que va dirigido.

Los alumnos después de haber creado una cuenta en el sistema deberán inscribirse a una clase para acceder a los temarios y ejercicios, en el área de trabajo, en esta los alumnos implementarán la lógica necesaria para dar solución a los casos de prueba a los que será sometido el ejercicio con el fin de evaluar los conocimientos en el área de programación.

Otra de las características del sistema es permitir ver el tiempo que el alumno ha invertido para dar solución a cada ejercicio, cuantas veces fue necesario mandar a calificar el programa para que este fuese aprobado exitosamente, sin superar los tiempos estipulados por el profesor, estos avances se podrán visualizar de dos formas la primera en gráficas de manera general y la segunda será más específica mediante reportes, dónde se detallará el tiempo, intentos y si el ejercicio fue resulto exitosamente. Los reportes en PDF los encontramos en las siguientes categorías para una mayor comodidad a la hora de evaluar a los alumnos.

Es por eso que ambas partes del sistema trabajan en conjunto para impulsar el ambiente distribuido sin dejar atrás la portabilidad, es así como se cumplen con los objetivos proporcionando ventajas a los alumnos ya que permite realizar consultas mediante los servicios disponibles.

## Trabajos futuros.

Como trabajos futuros se mencionan las siguientes mejoras:

* Implementar del sistema en una institución educativa real que requiera el servicio que SEA ofrece.
* Importar clases de java para agregar más funcionalidades en el sistema y ampliar la gama de ejercicios planteados.
* Agregar más tipos de dato a los métodos para cubrir todas las áreas necesarias para aprender a programar.

APÉNDICE A

Script de base de datos del Sistema.

1. Script de base de datos para el sistema SEA.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| componentes | Script. | | | |
| Función | Creación de tablas e inserción de registros para el sistema SEA. | | | |
| Autor | Alfonso Barrera García. | | | |
| Archivo | Ruta | Descripción | Fecha | Caso de uso |
| DB-SEAv1.0.sql | \Base\_datos\Script\_base\_datos\DB-SEAv1.0 | Permite la creación de tablas e inserción de registros, que serán de utilidad para el sistema SEA. | 15/08/2014 |  |

Tabla Script de base datos SEA.

APÉNDICE B

CODIGO FUENTE

1. Código fuente del Sistema SISE

B.1.1 Beans

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Componente** | Beans. | | | |
| **Función** | Se crearon los Bean con la finalidad de encapsular los datos de negocio, así se utiliza una única llamada a un método para enviar y recuperar los datos. | | | |
| **Autor** | Alfonso Barrera García. | | | |
| **Clase** | **Ruta** | **Descripción** | **Fecha** | **Caso de uso** |
| IndexBean.java | \Project\_SEA\src\java\mx\uatx\tesis\Beans\IndexBean.java | Encapsula los datos para mantener una comunicación entre los MB y DAO. | 10/10/2014. | 1. SEA-CU-001-00 Crear Cuenta. 2. SEA-CU-002-00 Recuperar Cuenta. 3. SEA-CU-003-00 Administrar Sesiones. |
| classProBean.java | \Project\_SEA\src\java\mx\uatx\tesis\Beans\classProBean.java | Encapsula los datos para mantener una comunicación entre los MB y DAO. | 10/10/2014 | 1. SEA-CU-004-00 Administrar Clases. |
| TemarioProBean.java | \Project\_SEA\src\java\mx\uatx\tesis\Beans\TemarioProBean.java | Encapsula los datos para mantener una comunicación entre los MB y DAO. | 05/01/2014 | 1. SEA-CU-005-00 Administrar Temarios. |
| BuiltExerciseProBean.java | \Project\_SEA\src\java\mx\uatx\tesis\Beans\BuiltExerciseProBean.java | Encapsula los datos para mantener una comunicación entre los MB y DAO. | 05/01/2014 | 1. SEA-CU-006-00 Administrar Ejercicios. |
| CasoPruebaProBean.java | \Project\_SEA\src\java\mx\uatx\tesis\Beans\CasoPruebaProBean.java | Encapsula los datos para mantener una comunicación entre los MB y DAO. | 05/01/2014 | 1. SEA-CU-006-00 Administrar Ejercicios. |
| ProfesorReporteBean.java | \Project\_SEA\src\java\mx\uatx\tesis\Beans\ProfesorReporteBean.java | Encapsula los datos para mantener una comunicación entre los MB y DAO. | 05/01/2014 | 1. SEA-CU-006-00 Administrar Avances. |
| AlumnoClassBean.java | \Project\_SEA\src\java\mx\uatx\tesis\Beans\AlumnoClassBean.java | Encapsula los datos para mantener una comunicación entre los MB y DAO. | 07/01/2014 | 1. SEA-CU-007-00 Inscribirse a una clase. |
| AlumnoTemarioBean.java | \Project\_SEA\src\java\mx\uatx\tesis\Beans\AlumnoTemarioBean.java | Encapsula los datos para mantener una comunicación entre los MB y DAO. | 07/01/2014 | 1. SEA-CU-008-00 Resolver Ejercicios. |
| AlumnoWorkSpaceBean.java | \Project\_SEA\src\java\mx\uatx\tesis\Beans\AlumnoWorkSpaceBean.java | Encapsula los datos para mantener una comunicación entre los MB y DAO. | 09/01/2014 | 1. SEA-CU-008-00 Resolver Ejercicios. |
| AlumnoReporteBean.java | \Project\_SEA\src\java\mx\uatx\tesis\Beans\AlumnoReporteBean.java | Encapsula los datos para mantener una comunicación entre los MB y DAO. | 09/01/2014 | 1. SEA-CU-009-00 Administrar Avances. |

Tabla Bean del sistema SEA.

B.1.2 DAO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Componente** | DAO. | | | |
| **Función** | Se crearon los DAO con la finalidad de realizar las operaciones básicas sobre las entidades que componen la aplicación. La funcionalidad ofrecida por este conjunto corresponde de un modo genérico con el patrón Data Access Object. | | | |
| **Autor** | Alfonso Barrera García. | | | |
| **Clase** | **Ruta** | **Descripción** | **Fecha** | **Caso de uso** |
| IndexDao.java | \Project\_SEA\src\java\mx\uatx\tesis\daos\IndexDao.java | Realiza las operaciones Insert, Delete, Select y Update a la base de datos. | 10/10/2014. | 1. SEA-CU-001-00 Crear Cuenta. 2. SEA-CU-002-00 Recuperar Cuenta. 3. SEA-CU-003-00 Administrar Sesiones. |
| classProDao.java | \Project\_SEA\src\java\mx\uatx\tesis\ daos \classProDao.java | Realiza las operaciones Insert, Delete, Select y Update a la base de datos. | 10/10/2014 | 1. SEA-CU-004-00 Administrar Clases. |
| TemarioProDao.java | \Project\_SEA\src\java\mx\uatx\tesis\ daos \TemarioProDao.java | Realiza las operaciones Insert, Delete, Select y Update a la base de datos. | 05/01/2014 | 1. SEA-CU-005-00 Administrar Temarios. |
| BuiltExerciseProDao.java | \Project\_SEA\src\java\mx\uatx\tesis\ daos \BuiltExerciseProDao.java | Realiza las operaciones Insert, Delete, Select y Update a la base de datos. | 05/01/2014 | 1. SEA-CU-006-00 Administrar Ejercicios. |
| CasoPruebaProDao.java | \Project\_SEA\src\java\mx\uatx\tesis\ daos \CasoPruebaProDao.java | Realiza las operaciones Insert, Delete, Select y Update a la base de datos. | 05/01/2014 | 1. SEA-CU-006-00 Administrar Ejercicios. |
| ProfesorReporteDao.java | \Project\_SEA\src\java\mx\uatx\tesis\ daos \ProfesorReporteDao.java | Realiza las operaciones Insert, Delete, Select y Update a la base de datos. | 05/01/2014 | 1. SEA-CU-006-00 Administrar Avances. |
| AlumnoClassDao.java | \Project\_SEA\src\java\mx\uatx\tesis\ daos \AlumnoClassDao.java | Realiza las operaciones Insert, Delete, Select y Update a la base de datos. | 07/01/2014 | 1. SEA-CU-007-00 Inscribirse a una clase. |
| AlumnoTemarioDao.java | \Project\_SEA\src\java\mx\uatx\tesis\ daos \AlumnoTemarioDao.java | Realiza las operaciones Insert, Delete, Select y Update a la base de datos. | 07/01/2014 | 1. SEA-CU-008-00 Resolver Ejercicios. |
| AlumnoWorkSpaceDao.java | \Project\_SEA\src\java\mx\uatx\tesis\daos \AlumnoWorkSpaceDao.java | Realiza las operaciones Insert, Delete, Select y Update a la base de datos. | 09/01/2014 | 1. SEA-CU-008-00 Resolver Ejercicios. |
| AlumnoReporteDao.java | \Project\_SEA\src\java\mx\uatx\tesis\ daos \AlumnoReporteDao.java | Realiza las operaciones Insert, Delete, Select y Update a la base de datos. | 09/01/2014 | 1. SEA-CU-009-00 Administrar Avances. |

Tabla Dao del sistema SEA.

B.1.3 Managed Bean

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Componente** | Managed Bean (MB). | | | |
| **Función** | Se crearon los MB con la finalidad de realizar operaciones de mayor complejidad que los paquetes DAO, responsabilizándose de coordinar las invocaciones de otros EJBs. | | | |
| **Autor** | Alfonso Barrera García. | | | |
| **Clase** | **Ruta** | **Descripción** | **Fecha** | **Caso de uso** |
| IndexMB.java | \Project\_SEA\src\java\mx\uatx\tesis\ManagedBean\IndexMB.java | Realiza las operaciones necesarias para la funcionalidad de login, recuperar contraseña y crear una nueva cuenta. | 10/10/2014. | 1. SEA-CU-001-00 Crear Cuenta. 2. SEA-CU-002-00 Recuperar Cuenta. 3. SEA-CU-003-00 Administrar Sesiones. |
| classProMB.java | \Project\_SEA\src\java\mx\uatx\tesis\ManagedBean\classProMB.java | Realiza las operaciones necesarias para la funcionalidad de administrar clases. | 10/10/2014 | 1. SEA-CU-004-00 Administrar Clases. |
| TemarioPrMB.java | \Project\_SEA\src\java\mx\uatx\tesis\ManagedBean\TemarioProMB.java | Realiza las operaciones necesarias para la funcionalidad de administrar los temarios. | 05/01/2014 | 1. SEA-CU-005-00 Administrar Temarios. |
| BuiltExerciseProMB.java | \Project\_SEA\src\java\mx\uatx\tesis\ManagedBean\BuiltExerciseProMB.java | Realiza las operaciones necesarias para la funcionalidad de administrar ejercicios. | 05/01/2014 | 1. SEA-CU-006-00 Administrar Ejercicios. |
| CasoPruebaProMB.java | \Project\_SEA\src\java\mx\uatx\tesis\ManagedBean\CasoPruebaProMB.java | Realiza las operaciones necesarias para la funcionalidad de administrar ejercicios. | 05/01/2014 | 1. SEA-CU-006-00 Administrar Ejercicios. |
| ProfesorReporteMB.java | \Project\_SEA\src\java\mx\uatx\tesis\ManagedBean\ProfesorReporteMB.java | Realiza las operaciones necesarias para la funcionalidad de generar reportes y gráficas. | 05/01/2014 | 1. SEA-CU-006-00 Administrar Avances. |
| AlumnoClassMB.java | \Project\_SEA\src\java\mx\uatx\tesis\ManagedBean\AlumnoClassMB.java | Realiza las operaciones necesarias para la funcionalidad de inscribirse a una clase. | 07/01/2014 | 1. SEA-CU-007-00 Inscribirse a una clase. |
| AlumnoTemarioMB.java | \Project\_SEA\src\java\mx\uatx\tesis\ManagedBean\AlumnoTemarioMB.java | Realiza las operaciones necesarias para la funcionalidad de listar los ejercicios. | 07/01/2014 | 1. SEA-CU-008-00 Resolver Ejercicios. |
| AlumnoWorkSpaceMB.java | \Project\_SEA\src\java\mx\uatx\tesis\ManagedBean\AlumnoWorkSpaceMB.java | Realiza las operaciones necesarias para la funcionalidad de dar solución a los ejercicios. | 09/01/2014 | 1. SEA-CU-008-00 Resolver Ejercicios. |
| AlumnoReporteMB.java | \Project\_SEA\src\java\mx\uatx\tesis\ManagedBean\AlumnoReporteMB.java | Realiza las operaciones necesarias para la funcionalidad de generar reportes y gráficas. | 09/01/2014 | 1. SEA-CU-009-00 Administrar Avances. |

Tabla Managed Bean del sistema SEA.

B.1.4 Vistas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Componente** | Vistas | | | |
| **Función** | Se crearon las vistas con la finalidad de tener una interfaz que le permita a los usuarios finales poder interactuar con el sistema SEA. | | | |
| **Autor** | Alfonso Barrera García | | | |
| **Clase** | **Ruta** | **Descripción** | **Fecha** | **Caso de uso** |
| Index.xhtml | \Project\_SEA\web\ Index.xhtml | Muestra los campos necesarios para el inicio de sesión, recuperar cuenta y de registrarse. | 10/10/2014. | 1. SEA-CU-001-00 Crear Cuenta. 2. SEA-CU-002-00 Recuperar Cuenta 3. SEA-CU-003-00 Administrar Sesiones. |
| Profesor\_Clases.xhtml | \Project\_SEA\web\ Profesor\_Clases.xhtml | Muestra los campos necesarios para agregar una clase, listar o eliminar. | 10/10/2014 | 1. SEA-CU-004-00 Administrar Clases. |
| Profesor\_Temario.xhtml | \Project\_SEA\web\ Profesor\_Temario.xhtml | Muestra los campos necesarios para agregar, listar o eliminar un tema o subtema. | 05/01/2014 | 1. SEA-CU-005-00 Administrar Temarios. |
| Profesor\_AgregarEjercicio. xhtml | \Project\_SEA\web\ Profesor\_AgregarEjercicio. xhtml | Muestra los campos necesarios para agregar, listar o eliminar un Ejercicio. | 05/01/2014 | 1. SEA-CU-006-00 Administrar Ejercicios. |
| Profesor\_AgregarCasoPrueba.xhtml | \Project\_SEA\web\ Profesor\_AgregarCasoPrueba.xhtml | Muestra los campos necesarios para agregar, listar o eliminar un caso de prueba con sus parámetros. | 05/01/2014 | 1. SEA-CU-006-00 Administrar Ejercicios. |
| Profesor\_graficasAvance. xhtml | \Project\_SEA\web\ Profesor\_graficasAvance. xhtml | Genera automáticamente las gráficas de acuerdo a los tiempos establecidos, al igual que los reportes en formato PDF con los detalles de cada ejercicio. | 05/01/2014 | 1. SEA-CU-006-00 Administrar Avances. |
| Alumno\_MisClases. xhtml | \Project\_SEA\web\ Alumno\_MisClases. xhtml | Lista las clases de cada profesor para que el alumno se pueda inscribir. | 07/01/2014 | 1. SEA-CU-007-00 Inscribirse a una clase. |
| Alumno\_Temario. xhtml | \Project\_SEA\web\ Alumno\_Temario. xhtml | Lista el temario correspondiente de la clase que se ha ingresado. | 07/01/2014 | 1. SEA-CU-008-00 Resolver Ejercicios. |
| Alumno\_WorkSpace. xhtml | \Project\_SEA\web\ Alumno\_WorkSpace. xhtml | Muestra los detalles de cada ejercicio y permite escribir líneas de código para compilar el ejercicio y calificar. | 09/01/2014 | 1. SEA-CU-008-00 Resolver Ejercicios. |
| Alumno\_graficasAvance. xhtml | \Project\_SEA\web\ Alumno\_graficasAvance. xhtml | Genera reportes detallados en formato PDF de los avances y una gráfica general. | 09/01/2014 | 1. SEA-CU-009-00 Administrar Avances. |

Tabla Vistas del sistema SEA.

# Bibliografía

Aguilar, L. J., & Zahonero Martínez, I. (2002). *Programación en Java 2.* Aravaca, Madrid, España: McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S.A. U. Recuperado el 10 de Febrero de 2013

Barahona, J. G., Seoane Pascual, J., & Robles, G. (2003). *Introducción al software libre.* Eureca Media, SL.

Bodie, L. (2005). *Mobile Messaging, Technologies and Services SMS, EMS, and MMS .* Wiley & Sons, Ltd. 2da. ed.

Booch, G., Rumbaugh, J., & Jacobson, I. (2003). *EL LENGUAJE UNIFICADO DE MODELADO.* España: FARESO, S.A. .

CodingBat. (14 de Mayo de 2015). Obtenido de codingbat: http://codingbat.com/

Díaz, G. R. (10 de Mayo de 2004). *UDLAP BIBLIOTECAS.* Obtenido de http://catarina.udlap.mx/u\_dl\_a/tales/documentos/lis/rivera\_d\_g/capitulo\_8.html#

Eclipse Foundaton, I. (15 de Noviembre de 2004). *Eclipse Foundation*. Obtenido de www.eclipse.org

Elizalde, J. A. (2009). *Sistema de pedido a través de telefonos celulares con tecnología JAVA ME.* Calpulalpan, Tlaxcala.

Kroenke, D. M. (2003). *Procesamiento de bases de datos.* Mexico: Pearson Educación.

Molina, F. J. (2004). *Redes de Área Local.* Madrid, España: ALFAOMEGA GRUPO EDITOR, S.A. de C.V.

Newton, M., Cobb, J., Weaver, C., Kozmik, R., Krzyzanek, L., Vlastimil Elias, . . . Hladky, D. (s.f.). *JBoss Community*. Obtenido de JBoss Application Server: http://www.jboss.org/jbossas/downloads/

Oracle. (17 de Mayo de 2014). *java.sun.com*. Obtenido de java.sun.com: http://java.sun.com/docs/books/tutorial/2e/book.html

Pedro, A. A. (1983). Principios y procedimientos de evaluación tradicional. En A. A. Pedro, *Principios y procedimientos de evaluación tradicional* (págs. Pág. 13 -37).

Programming Language Popularity. (25 de Octubre de 2013). *Programming Language Popularity*. Obtenido de Programming Language Popularity: http://web.archive.org/web/20150829085248/http://langpop.com/

Programming Language Popularity. (25 de 10 de 2013). *Programming Language Popularity*. Obtenido de Programming Language Popularity: http://web.archive.org/web/20150829085248/http://langpop.com/

Ramírez, R. S. (Marzo de 2006). *Sistema de reuniones inalambrico para dispositivos móviles con tecnologia Bluetooth*. Obtenido de http://mixteco.utm.mx/~resdi/historial/Tesis/Tesis\_Rubi.pdf

Real Academia Española. (23 de Julio de 2015). Obtenido de Real Academia Española: http://dle.rae.es/?w=evaluaci%C3%B3n&o=h

tutorialspoint. (23 de Agosto de 2015). *tutorialspoint*. Obtenido de tutorialspoint: http://www.tutorialspoint.com/

Velázquez, J. L. (2010). *Lenguajes de programación: Introduccion a C/C++ IDE´S.* Guanajuato.

wikipedia. (22 de 01 de 2001). *wikipedia*. Obtenido de wikipedia: http://es.wikipedia.org/wiki/Evaluaci%C3%B3n

wikipedia. (29 de Abril de 2015). Obtenido de wikipedia: https://es.wikipedia.org/wiki/Lenguaje\_unificado\_de\_modelado

*wikipedia*. (17 de Julio de 2015). Obtenido de Modelo–vista–controlador: https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador

*Xampp*. (s.f.). Obtenido de Apache Friends: http://www.apachefriends.org/es/xampp.html