

**Universidad de las Ciencias Informáticas**  
**Facultad 6**



**Sistema de gestión para el control de acceso y los servicios  
prestados del Centro de Idiomas Extranjeros de la Universidad de  
las Ciencias Informáticas.**

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

**Autor:** Sayuri Gutiérrez Mena

**Tutor:** Ing. Luanner Kerton Martínez

**La Habana, 2016**

**“Año 58 de la Revolución”**

**DECLARACIÓN DE AUTORÍA**

Declaración de autoría

Declaro ser autor de la presente tesis y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio.

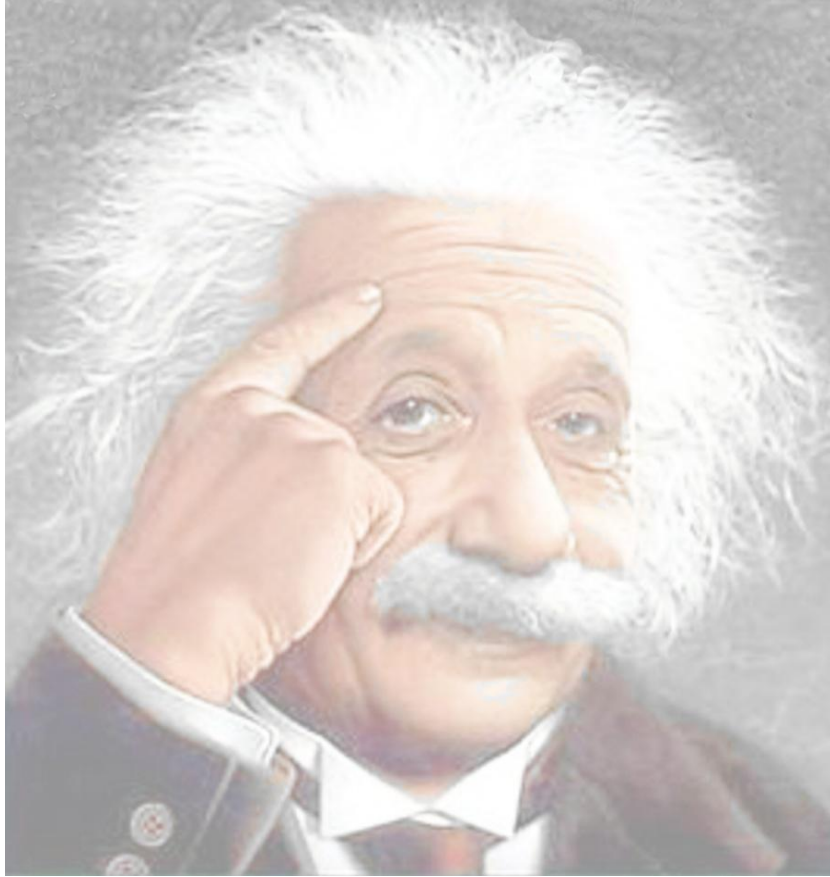
Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Sayuri Gutiérrez Mena

Ing. Luanner Kerton Martínez

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Tutor



"Si buscas resultados distintos, no hagas siempre lo mismo."

*A. Einstein*

Autor:

- ✓ Sayuri Gutiérrez Mena
- ✓ Universidad de las Ciencias Informáticas.
- ✓ e-mail: sayuri@uci.cu

Tutor:

- ✓ Ing. Luanner Kerton Martinez
- ✓ Universidad de las Ciencias Informáticas.
- ✓ e-mail: luanner@uci.cu

Graduado de Ingeniero en Ciencias Informáticas. Ha trabajado como líder, analista y desarrollador de sistemas de gestión de información y ha colaborado en la implantación de Sistemas de Gestión Integrada bajo normas ISO. Tiene registrada la participación en eventos y publicaciones con sistemas de información para escenarios agrícolas, así como herramientas para la comunicación interna en redes de negocios. Actualmente trabaja como desarrollador del proyecto SCADA del Centro de Informática Industrial de la Universidad de las Ciencias Informáticas. Cursa la maestría de Informática Aplicada donde investiga sobre la aplicación de los repositorios de instancias de procesos en empresas de servicios.

Dedicatoria

*Dedico de manera especial a mis Padres Roberto y Zelma por ser el principal cimiento para la construcción de mi vida profesional, sentaron en mí las bases de responsabilidad y deseos de superación, en ustedes tengo el espejo en el cual me he reflejado, pues sus virtudes infinitas y su gran amor me han llevado a admirarlos cada día más.*

*A mi hermana, abuelos, esposo y familia quienes me han brindado su comprensión, cariño, amor y confianza en que iba a cumplir con mis ideales.*

## Agradecimientos

*Quiero agradecer de corazón a mis excelentes padres por transmitirme su ejemplo, por su apoyo incondicional, su amor, su esfuerzo y dedicación en cada momento de mi vida. Ustedes son la fuente de mi inspiración y mis ganas de salir adelante. Gracias por todo mamá y papá.*

*Papa sé que si estuvieras conmigo me hubieras ayudado como lo hizo mamá en esta etapa tan importante de mi vida como siempre lo hiciste. Sé que estarías muy orgulloso de mi persona una vez más como lo está mama. Por siempre estarás en mi corazón.*

*A mi hermana gracias por cada momento vivido durante todos estos años y los q nos faltan por vivir, por estar siempre pendiente de mí y apoyándome. Nuestros corazones siempre están y estarán unidos por el amor de hermanas.*

*A mis abuelos aunque no estén conmigo físicamente están en mi corazón en todo momento, sé que les hubiese gustado verme en estos momentos y seguro estuvieran orgullosos de mí.*

*A mi madrina muchas gracias por tu preocupación, por tu apoyo incondicional y por tu confianza en mí hacia la tesis y en todos los momentos de mi vida. Eres única, eres un regalo muy valioso que tengo y tendré por siempre.*

*A mi esposo Daniel gracias por ayudarme en el momento más difícil de mi carrera ,por estar siempre a mi lado, por tus palabras, por acompañarme siempre, por tu confianza y por todo el amor que me brindas día a día. Me ayudaste hasta donde te era posible, incluso más que eso.*

*A la familia de mi esposo que también es la mía, muchas gracias por estar siempre presente, apoyándome en todo momento y por trasmitirme su amor y su cariño día a día.*

*A ti Jorgito más que primo has sido como un hermano para mí, gracias por estar siempre presente en todos los momentos de mi vida.*

*A toda la familia gracias por siempre estar pendiente de mi persona.*

*A mi querida amiga Yohanis que ha sido mi brazo derecho en todo momento, mi pañuelo de lágrimas, mi consejera, gracias por apoyarme sobre todo en los momentos más difíciles de mi vida.*

*A Liliana por siempre brindarme su apoyo durante estos 10 años, gracias por ser como una madre para mí en la UCI.*

*A todas las personas que quiero y admiro a Yordan, Nelson, Lili, Osvaldo, Rene, Lázaro, Yusle, Edelis, Elizabeth gracias por apoyarme y por estar pendientes de mí en el transcurso de mi vida y de mi carrera.*

*A mi tutor por guiarme en la realización de este trabajo, gracias por su tiempo, ayuda y comprensión.*

*A los profesores del tribunal y al oponente muchas gracias por su dedicación, criterio y por transmitirme sus conocimientos.*

*A mis compañeros de trabajo gracias por brindarme su apoyo en todo.*

*A la Universidad que nos abrió sus puertas para ser mejores personas y buenos profesionales.*

*A mis compañeros de aula Yaima, Yadriel, Pepe, Yoandri, Eduardo, Yudelsy, Marydunia y Yamil por compartir todos estos años de Universidad, ustedes más que compañeros fueron mis amigos, gracias.*

*A todos mis profesores gracias por haberme enseñado a ser mejor en la vida y a realizarme profesionalmente.*

*A todas aquellas personas que durante mi carrera estuvieron a mi lado apoyándome.*

*Muchas gracias a todos.*

## Resumen

Los centros de idiomas constituyen casas de estudio para la superación profesional e intelectual tanto en centros de trabajo, educativos y otras personas interesadas. La gestión de información en dichos centros determina en gran parte, su eficiencia y eficacia e incide en la calidad del proceso de enseñanza y aprendizaje. La Universidad de Ciencias Informáticas, a partir del nuevo modelo educativo, ha constituido su centro de idiomas uniendo en sus servicios aquellos brindados tradicionalmente por los Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros (CASIE). Actualmente, la gestión de información en los CASIE es llevada a cabo por los técnicos generales en registros de Microsoft Office Excel, método insuficiente para obtener información sobre la disponibilidad de espacios, actividades, servicios y préstamos, lo que afecta la planificación, evaluación y toma de decisiones. En la presente investigación se evidencia el desarrollo de un sistema de gestión para el control de acceso y servicios brindado por los CASIE del Centro de Idiomas de la Universidad de las Ciencias Informáticas. El sistema es una plataforma web que permitirá integrar la gestión de información, desarrollado en Symfony 2 utilizando la metodología *OpenUp*.

Palabras claves: acceso, centro de idiomas, CASIE, gestión de información, servicios.

*Abstract*

*Language centers are houses of study for professional and intellectual improvement both in workplaces, educational and other interested persons. Information management in these centers largely determines its efficiency and effectiveness and affects the quality of teaching and learning. University of Computer Science, from the new educational model has been its language center uniting in those services traditionally provided by the Centers for independent learning and Foreign Language Services (CASIE). Currently, the management of information in CASIE is carried out by general technicians records Microsoft Office Excel, insufficient to obtain information on the availability of spaces, activities, services and lending method, which affects planning, evaluation and decision making. In this research the development of a management system for access control and CASIE provided by the Language Center of the University of Information Science services is evident. The system is a web platform that will integrate information management, developed in Symfony 2 using OpenUP.*

*Keywords: access, language centers, CASIE, information management services.*



## Contenido

<b>Resumen .....</b>	<b>VII</b>
<b>Introducción .....</b>	<b>1</b>
<b>Capítulo 1 Fundamentación Teórica.....</b>	<b>5</b>
1.1 Marco Conceptual.....	5
1.1.1 Centros de idiomas.....	5
1.1.2 Servicios de centros de idiomas .....	5
1.1.3 Proceso de control de acceso a salones de aprendizaje.....	6
1.1.4 Gestión de información en centros de idiomas .....	7
1.2 Sistemas automatizados relacionados con el campo de acción.....	8
1.2.1 Proyecto Open Marco Polo .....	8
1.2.2 Koha UNPL.....	9
1.2.3 Sistema de gestión de estudios de la universidad de Madrid.....	9
1.2.4 Langlish .....	9
1.2.5 Sistema Recumat .....	9
1.3 Tecnologías y herramientas de desarrollo .....	10
1.3.1 Elección de la metodología de desarrollo de software .....	10
1.3.2 Lenguaje de modelado .....	13
1.3.3 Lenguaje de programación .....	15
1.3.4 Elección del framework de desarrollo .....	16
1.3.5 Sistema gestor de Base de Datos.....	17
1.3.6 Herramientas de apoyo para el desarrollo de software .....	18
1.4 Conclusiones parciales .....	21
<b>Capítulo 2 Análisis y diseño del sistema .....</b>	<b>22</b>
2.1 Modelo de Negocio.....	22
2.1.1 Descripción del negocio.....	22

2.1.2	Reglas del negocio a considerar.....	22
2.1.3	Actores del negocio .....	22
2.1.4	Diagrama de casos de uso del negocio .....	22
2.1.5	Descripción de casos de uso del negocio .....	23
2.1.6	Propuesta de solución. ....	24
2.2	Requisitos de software.....	24
2.2.1	Requisitos funcionales.....	24
2.2.2	Requisitos no funcionales .....	26
2.2.3	Actores del sistema .....	27
2.2.4	Diagrama de casos de uso del sistema .....	27
2.2.5	Especificaciones de caso de uso .....	28
2.3	Diseño de la solución.....	35
2.3.1	Arquitectura .....	35
2.3.2	Patrón de diseño.....	36
2.3.3	Diagrama de clases de diseño.....	38
2.3.4	Modelo de datos .....	39
2.3.5	Diagrama de despliegue .....	41
2.4	Conclusiones parciales .....	42
<b>Capítulo 3 Implementación y pruebas.....</b>		<b>43</b>
3.1	Modelo de implementación .....	43
3.1.1	Tratamiento de errores .....	43
3.1.2	Estándar de codificación.....	44
3.1.3	Diagrama de componentes.....	44
3.2	Pruebas de software .....	45
3.2.1	Pruebas para el caso de uso Autenticar Usuario .....	46
3.2.2	Pruebas para el caso de uso gestionar Recursos .....	47

3.2.3	Pruebas para el caso de uso Gestionar préstamos.....	48
3.2.4	Pruebas para requisitos no funcionales .....	50
3.2.5	Validación de resultados obtenidos.....	51
3.3	Conclusiones parciales .....	54
	<b>Conclusiones .....</b>	<b>55</b>
	<b>Recomendaciones .....</b>	<b>56</b>
	<b>Citas.....</b>	<b>57</b>
	<b>Anexos.....</b>	<b>63</b>
	<i>Anexo no. 1: Encuesta .....</i>	<i>63</i>
	<i>Anexo no. 2: Interfaz Autenticar usuario.....</i>	<i>63</i>
	<i>Anexo no. 3: Interfaz Registrar un Préstamo .....</i>	<i>64</i>

**Índice de tablas**

Tabla 1.	Diagrama de actividades caso de uso Controlar Acceso .....	23
Tabla 2.	Diagrama de actividades caso de uso Gestionar Préstamos .....	23
Tabla 3.	Actores del sistema .....	27
Tabla 4.	Especificación Caso de uso Autenticar usuario .....	30
Tabla 5.	Especificación Caso de uso Gestionar préstamo.....	33
Tabla 6.	Especificación Caso de uso Gestionar reservaciones .....	35
Tabla 7.	Descripción de las entidades de la base de datos. ....	41
Tabla 8.	Secciones a probar Caso de uso Autenticar usuario .....	46
Tabla 9.	Matriz de pruebas de datos caso de uso autenticar usuario .....	47
Tabla 10.	Secciones a probar Caso de uso Gestionar recurso .....	47
Tabla 11.	Variables del caso de Uso Gestionar recurso .....	48
Tabla 12.	Matriz de pruebas de datos .....	48
Tabla 13.	Secciones a probar Caso de uso Gestionar préstamo .....	49
Tabla 14.	Variables del caso de Uso Gestionar préstamo .....	49
Tabla 15.	Matriz de pruebas de datos Gestionar préstamo.....	50
Tabla 16.	Resultados de la prueba con el JMeter .....	51
Tabla 17.	Criterios para la validación de la solución por el usuario.....	52

**Índice de figuras**

Figura 1. Fases de la metodología RUP .....	12
Figura 2. Capas de OpenUP .....	13
Figura 3. Diagrama de casos de uso del negocio.....	23
Figura 4. Diagrama de casos de uso del sistema .....	28
Figura 5. Diagrama de paquetes Sistema .....	36
Figura 6. Vista del patrón Decorador.....	38
Figura 7. Diagrama de clases Caso de uso Gestionar préstamos.....	39
Figura 8. Diagrama de clases Caso de Uso Gestionar Reservación .....	39
Figura 9. Diagrama Entidad-Relación.....	40
Figura 10. Diagrama de despliegue .....	42
Figura 11. Validación de datos de entrada en las vistas .....	43
Figura 12. Diagrama de componentes.....	45
Figura 13. Registro de No Conformidades.....	46
Figura 14. Variables del caso de Uso Autenticar Usuario .....	47
Figura 15. Resultados de prueba con JMeter .....	51
Figura 16. Registro de acceso de personas.....	52
Figura 17. Registro de reservaciones .....	53
Figura 18. Informe de acceso por actividad .....	53

## Introducción

El mundo actual presenta un desarrollo vertiginoso, que obliga a aplicar nuevos métodos de gestión, producción y servicios. La tendencia muestra que las nuevas sociedades exigen personal capacitado para afrontar los constantes cambios del medio ambiente, por tanto, los modelos y métodos de enseñanza se perfeccionan para obtener mejores profesionales.

Cuba ha sido reconocida por diversas organizaciones internacionales como una vanguardia educativa, mostrando altos indicadores en la calidad de enseñanza, la innovación y el alcance de su modelo educacional. Como parte de la estrategia para informatizar Cuba, se crea en el 2002 la Universidad de las Ciencias Informáticas (UCI), que incluye en su plan de estudios un conjunto de asignaturas que permiten la formación de profesionales aptos en informática y comprometidos con la sociedad.

Uno de los perfiles necesarios en la carrera es el dominio de lenguas extranjera, precisamente el inglés, gracias a la relevancia que tiene dentro de la rama, está incluido en el plan de estudios. Como estrategia para propiciar el aprendizaje de idiomas se crean los Centros de Autoaprendizaje y Servicios de Idiomas Extranjeros (CASIE). La universidad cuenta con cuatro CASIE de forma que garantiza el acceso a estos por profesores, especialistas, cursantes e interesados. Los CASIE promueven el aprendizaje de lenguas extranjeras ofreciendo asistencia, recursos y espacios especializados; y cuentan con los siguientes espacios: un salón de lectura, uno de audición y una sala de computación.

En cada CASIE labora un técnico general y un técnico informático. El técnico general entre otras funciones lleva el control de acceso a los distintos salones. Además, deben detallar los servicios solicitados, el nombre de quien accede, si es estudiante o asesor, los préstamos, entre otros datos.

El nuevo modelo de enseñanza en la Universidad define la creación de un Centro de Idiomas (CI) que agrupará a los 4 CASIE. El mismo ampliará las funciones de los CASIE y se comportará como los otros centros de idiomas del país. Actualmente el CI no cuenta con la base de información necesaria para caracterizar de forma precisa el uso de los servicios que brindan los CASIE, específicamente, no tiene ningún método fiable para evaluar el acceso y los recursos demandados.

La Universidad ha realizado intentos de automatizar la función en los CASIE, como el caso del sistema Recumat, desarrollado en el 2011 para el control de materiales y mejorado en el 2012 con mecanismos de seguridad y gestión de préstamos. No obstante, la herramienta no se utiliza por problemas técnicos de seguridad, mala definición de funcionalidades básicas y poca interacción con el usuario.

El control de la información de acceso físico y uso de servicios se llevan en una hoja de cálculo de Microsoft Office Excel, a través de tablas para los asesores, grupos y usuarios. Esto dificulta la capacidad

de los CASIE de tener estadísticas de acceso a los salones, control de los grupos, detalles de los visitantes y afecta la toma de decisiones, lo que propicia que no se contribuya de forma adecuada al mejoramiento del proceso docente educativo. El análisis de los CASIE arrojó los siguientes problemas:

1. Durante el análisis de información se pierde tiempo sintetizando los datos de las tablas para generar informes, los que son limitados y no brindan información suficiente para la toma de decisiones: Los informes se realizan a partir de datos obtenidos del Excel que llevan los técnicos, que no registran todas las informaciones necesarias.
2. Es difícil determinar la frecuencia con que asisten los asesores y estudiantes a los salones del centro. Para obtener mayor información sobre el uso de los servicios los informes deben incluir la frecuencia de acceso. Actualmente es difícil de determinar dado a que es necesario asociar los mismos parámetros en varios Excel.
3. Dado los problemas anteriores durante la planificación de actividades los trabajadores de los CASIE no pueden tener en cuenta los servicios de más uso según un tipo específico de usuario.

**Problema a resolver:** El Centro de Idiomas de la Universidad de las Ciencias Informáticas no tiene un control efectivo del acceso y uso de sus servicios, lo que afecta la toma de decisiones y no permite caracterizar el proceso a partir de datos reales.

**Objeto de estudio:** Gestión de información en los Centros de Idiomas.

**Campo de acción:** Proceso de gestión y control de información en el Centro de Idiomas de la Universidad de Ciencias Informáticas.

**Objetivo general:** Desarrollar un sistema de gestión para el control de acceso y servicios en el Centro de Idiomas de la Universidad de Ciencias Informáticas.

**Objetivos específicos:**

1. Generar una base de datos fiable sobre el acceso y uso de servicios en los CASIE que permita a la dirección del Centro de Idiomas realizar el análisis para la toma de decisiones.
2. Implementar una herramienta para la gestión de servicios en los CASIE de la Universidad de las Ciencias Informáticas que sea adaptable a los posibles cambios del nuevo modelo de enseñanza.

**Tareas de la Investigación:**

1. Análisis del proceso de control de acceso físico y servicios de salones de los CASIE.

2. Caracterización de las herramientas automatizadas para el control de acceso físico y gestión de servicios.
3. Captura de las necesidades de automatización dentro del proceso de control de acceso y servicios en los CASIE.
4. Selección de las herramientas y tecnologías a utilizar durante el desarrollo del sistema propuesto.
5. Análisis y Diseño de la solución a partir de las necesidades, tecnologías y herramientas identificadas.
6. Implementación de la solución propuesta
7. Validación de la solución.

Como **Posibles resultados**: Un sistema de gestión para los CASIE que permita la automatización del control de acceso y los servicios brindados por el Centro de Idiomas de la Universidad de las Ciencias Informáticas.

Desde el punto de vista metodológico se utilizan los siguientes métodos científicos:

#### **Métodos teóricos:**

**Analítico – Sintético:** Utilizado para analizar elementos bibliográficos y definiciones sobre los sistemas de información, control de acceso y gestión de servicios, con el objetivo de arribar a conclusiones que sustenten la necesidad de la investigación.

**Histórico – Lógico:** Se utilizará para evaluar evolución histórica del proceso de gestión de servicios en centros de aprendizaje, así como analizar las soluciones existentes y su evolución tecnológica, las tendencias más recientes y complementar las características que debe tener la aplicación

#### **Métodos empíricos:**

**Observación:** Se utilizará para observar el proceso de control de acceso en los CASIE, caracterizar el proceso desde su propio escenario y documentar los procedimientos observados

**Entrevista:** Se aplicará a especialistas y técnicos del CASIE antes y después de la solución para: obtener información sobre el proceso de gestión en los CI, identificar las necesidades de automatización, evaluar el impacto de la herramienta.

**Encuesta:** Se utilizará para obtener información que permita cuantificar el grado de automatización y satisfacción del personal beneficiado antes y después de la implantación de la herramienta. [Ver Anexo 1.](#)



El contenido en la investigación se estructurará de la siguiente manera:

### **Capítulo 1: Fundamentación Teórica**

En este capítulo se plantean los conceptos relacionados con la gestión de servicios y control de acceso físico. Se presentan los argumentos teóricos de la investigación y se definen las tecnologías y herramientas que serán utilizadas para el desarrollo del sistema.

### **Capítulo 2: Descripción del sistema**

Se detallará el proceso de control de los CASIE, se definirá el objeto de investigación y se levantarán los requisitos del sistema. Se describirá el diseño de la solución a través de los artefactos de la ingeniería seleccionada.

### **Capítulo 3: Implementación y validación de pruebas**

Se detallará la implementación del sistema, se expondrán las técnicas utilizadas para las pruebas, así como los resultados. Además, se reflejarán las conclusiones de la validación realizada.

## Capítulo 1 Fundamentación Teórica

### Introducción

En el presente capítulo se describen los principales conceptos relacionados con el control de acceso y los servicios que se brindan en los CASIE, así como los sistemas automatizados que se relacionan con el campo de acción de la investigación. Posteriormente, se definirán las tecnologías y herramientas para el desarrollo de la solución justificando en cada caso su utilización.

### 1.1 Marco Conceptual

#### 1.1.1 Centros de idiomas

Los centros de idiomas constituyen el reflejo de la voluntad de ampliar el dominio lingüístico de los profesionales que compiten dentro del mercado laboral. Ante la creciente demanda de personal con capacidad para comunicarse en varios idiomas se fundamentó dentro de los perfiles académicos, la necesidad de que los egresados (principalmente universitarios) tuvieran dominio de segundas y terceras lenguas. Actualmente, esta necesidad es un requisito indispensable en la mayoría de las disciplinas (1).

Así, se crearon espacios dedicados al desarrollo, entrenamiento y certificación de idiomas en las personas, y aunque inicialmente fue un espacio dedicado al perfil profesional y universitario, dado el aumento de mercado laboral y la globalización, en la mayoría de los países los centros de idiomas son entidades dirigidas a cualquier interesado.

Actualmente, la mayoría de las universidades consideradas como centros de altos de estudios tienen programas de idiomas dirigidos desde sus centros. Las actividades y servicios dentro de estos centros están muy relacionados con los tradicionales préstamos de recursos, como mismo pasa en las bibliotecas, e incluyen además servicios académicos y especializados en dependencia de su alcance, misión, idiomas y modelo de enseñanza (2).

#### 1.1.2 Servicios de centros de idiomas

Los servicios en general abarcan las siguientes asistencias (3):

1. Préstamos: Consiste en la facilitación al usuario de materiales y recursos, donde el usuario asume temporalmente la custodia de dicho medio. Los préstamos pueden ser:
  - a. Préstamo de sala: Consulta de los materiales en los salones de lectura o áreas designadas dentro de la biblioteca.

- b. Préstamo a domicilio: Los usuarios podrán llevarse los materiales a casa mediante una solicitud generalmente escrita y firmada por la entidad, que lo responsabiliza del medio adjuntado.
2. Maestros nativos: Los maestros nativos o maestros especializados en lenguas son parte esencial del funcionamiento de los centros de idiomas. Dichas personas preparan los programas, imparten los cursos y certifican las habilidades de los cursantes. Transmiten los idiomas de los que tienen dominio utilizando recursos pedagógicos, compartiendo los elementos culturales y facilitando el enriquecimiento del proceso educativo.
3. Aprendizaje fundamentado en metodologías reconocidas: Las metodologías reconocidas o de referencias contienen parte de la experiencia de modelos de éxito y certificados por instituciones o programas de prestigio. Los centros de idiomas adoptan dichas metodologías para ofrecer paquetes idiomáticos estandarizados que aumentan su prestigio, facilitan la comunicación con otros centros y aportan mayores materiales y medios de calidad. Como ejemplos de estas metodologías está *Oxford English Methodology*, una de las más utilizadas en los centros que enseñan el idioma inglés.
4. Apoyo didáctico soportado por tecnología de punta: Con la inclusión de las nuevas tecnologías de la comunicación y la informática además de las nuevas herramientas de enseñanza asistida por computadora y aprendizaje virtual, los centros de idiomas han adoptado las nuevas tecnologías para dar mayor soporte al proceso de transmisión de conocimientos. Programas orientados a diferentes edades, herramientas didácticas, sistemas de evaluación, aplicaciones en la nube, redes sociales, juegos, medios audiovisuales, entre otros, son parte de las nuevas tecnologías adoptadas.
5. Diseño de cursos para grupos especiales: Aunque los programas de idiomas son generales, con cursos preparados con antelación y contenido bien definidos, como método para acoger mayor cantidad de clientes y adaptarse a las necesidades de estos, los centros de idiomas tienen la capacidad de ofrecer cursos especializados según las necesidades de sus clientes. Los cursos especializados tienen en cuenta el idioma, país de origen, edad, conocimientos previos, cultura y objetivos de aprendizaje (4).

Estos son los principales servicios y características de la mayoría de las escuelas y centros de idiomas, de los cuales la mayor parte radica en las universidades, que son a su vez, quienes marcan las tendencias y los métodos de enseñanza, las tecnologías a utilizar e incluso, las reglas de acceso.

### **1.1.3 Proceso de control de acceso a salones de aprendizaje**

El acceso a los centros de idiomas se controla de distintas formas. Depende fundamentalmente del alcance del centro, su público objetivo, ubicación y las características de la sociedad. Dentro de las universidades, el acceso suele realizarse a través de solicitudes formales o según los programas de estudios de las carreras. Existen universidades y centros que incluyen entre sus normas de acceso conductas de vestuario. Desde el punto de vista del funcionamiento, el acceso generalmente se caracteriza por los siguientes aspectos:

1. El control de la actividad a realizar por las personas que acceden a los centros: Los centros tienen un área de registro donde las personas que acceden (ya sea mediante pase por matrícula o acceso libre) dejan información de la actividad a realizar, horario o referencia en caso de asistir mediante contrato previo con tarea o persona del centro.
2. Control de horarios: La mayoría de las salas tienen acceso normado mediante turnos u horarios previamente establecidos por el personal de planificación de los centros.

Estas dos tareas permiten que los centros de aprendizaje de idiomas tengan procesos de gestión de información.

#### **1.1.4 Gestión de información en centros de idiomas**

La Gestión de información (GI) es un área que suele confundirse con la gestión del conocimiento (GC). El concepto teórico de la gestión de información abarca el proceso de generación, coordinación, almacenamiento, conservación, búsqueda y recuperación de la información por parte de los interesados. Por tanto, su objetivo es asegurar la integridad, disponibilidad y confidencialidad de la información (5).

Dentro de un centro de idiomas, la información obtenida por el control de acceso y los servicios, es gestionada a partir de distintas herramientas. Dentro de los procesos de GI, la generación es la actividad generada inherentemente por el flujo documental de los centros de idiomas. Los sitios con infraestructura desarrollada, prácticas estandarizadas y sistemas de gestión automatizados, realizan exitosamente la coordinación, almacenamiento, recuperación y búsqueda. La realización exitosa de estas tareas, es la que supone la realización de la gestión documental, que especifica las técnicas que permiten manejar el flujo de documento para extraer conocimiento (6).

La introducción de las Tecnologías de la Información y las Comunicaciones ha intervenido en los procesos de GI, de tal modo que como parte del tratamiento de información, actualmente las exigencias se basan en sistemas automatizados para la gestión del conocimiento, que incluyen desde bases de datos hasta mecanismos inteligentes para un análisis de datos.

Con herramientas automatizadas, se puede cumplir con el concepto más general de la GI, que determina que es todo lo relacionado con la obtención de la información adecuada, en la forma correcta, para la persona indicada, al costo adecuado, en el tiempo oportuno, en el lugar apropiado, para tomar la acción correcta (7). Esta definición abarca los objetivos de la gestión de información, que son:

- Maximizar el valor y los beneficios derivados del uso de la información.
- Minimizar el costo de adquisición, procesamiento y uso de la información.
- Determinar responsabilidades para el uso efectivo, eficiente y económico de la información.
- Asegurar un suministro continuo de la información.

Las organizaciones son sistemas inteligentes, éstas deben ser generadoras, almacenadoras y transformadoras de los conocimientos que le permitan enfrentar y modificar su entorno, ya sea para adaptarse a él o, de ser posible, para adaptar el entorno a su beneficio. Es común afirmar que en ambientes tan complejos como los que deben enfrentar hoy en día las organizaciones, sólo aquellas que utilicen todos los medios a su alcance podrán lograr el objetivo de ser exitosas. Las tecnologías de información juegan un papel central en esta enloquecida carrera emprendida por las empresas contemporáneas.

## **1.2 Sistemas automatizados relacionados con el campo de acción**

Los Sistemas de Información representan un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad (objetivo). Dichos elementos formarán parte de alguna de estas categorías (8).

- Personas
- Datos
- Actividades o técnicas de trabajo.
- Recursos materiales (recursos informáticos)

Algunos de los sistemas automatizados utilizados para la gestión de información en centros de idiomas son:

### **1.2.1 Proyecto Open Marco Polo**

Es un proyecto de software libre creado con el objetivo de administrar las tareas internas y externas de una biblioteca, así como automatizar completamente las tareas relacionadas con el usuario. Trabaja íntegramente con bases de datos referencias como Isis y está programado para funcionar como un ambiente Web. Tiene requerimientos mínimos de instalación, facilidad de uso y sin limitaciones de

conexiones de usuario. A pesar de ser diseñado para biblioteca, sus funciones han sido extendidas a centros de idioma, fundamentalmente en Argentina. Su desarrollo modular ha permitido agregar módulos de gestión, administración, catálogos y préstamos (9).

### **1.2.2 *Koha UNPL***

Es un sistema integral para la gestión de conocimiento en centros bibliotecarios y de servicios académicos, desarrollado específicamente para la Universidad de La Plata. Fue desarrollado parcialmente con herramientas de software libre. Opera bajo sistema GNU/Linux. Permite realizar todos los procesos necesarios, que van desde la adquisición de los materiales hasta los servicios a usuarios (10).

### **1.2.3 *Sistema de gestión de estudios de la universidad de Madrid***

Es un sistema creado para la gestión de servicios bibliotecarios y centros de enseñanzas de idiomas vinculados con la Universidad de Madrid. Se creó como proyecto de fin de curso y su fundamental característica es que está diseñado para interfaces móviles. Tiene las capacidades gestión de préstamo, gestión de acceso, reservaciones, inventario de materiales, evaluaciones, entre otras características (11).

Estos sistemas presentan las siguientes características en común: Fueron desarrollados inicialmente para la gestión de bibliotecas y extendidos a procesos de centros de idiomas; están desarrollados de forma específica para un área determinada. Estas características determinan que, aunque se basan en herramientas libres, carecen de las funcionalidades necesarias y el alcance para utilizarlos en la resolución del problema en cuestión.

Por dicho motivo, se hizo necesario revisar las soluciones desarrolladas en la Universidad de las Ciencias Informáticas para la gestión de información de los CASIE.

### **1.2.4 *Langlish***

Es un sistema de apoyo a la autogestión del conocimiento en los CASIE. Su objetivo es propiciar un ambiente web para realizar diferentes diagnósticos a los estudiantes según su nivel y conocimientos previos, para luego analizar los resultados y apoyar la enseñanza personalizada. Ayuda en la confección de preguntas y su colección dentro de cuestionarios que pueden ser dirigidos a grupos específicos de usuarios (12).

### **1.2.5 *Sistema Recumat***

Es un sistema desarrollado para la gestión de recursos materiales y humanos de los CASIE, que fue ampliado en el 2012 para soportar también los procesos de gestión de préstamos. El sistema está basado en tecnología web y permite detallar los recursos existentes y realizar préstamos sobre estos (13).

Estos sistemas están desarrollados específicamente para los CASIE de la Universidad de las Ciencias Informáticas. No obstante, tienen las siguientes deficiencias:

Presentan obsoleto. Aunque están desarrollados para plataformas web no utilizan versiones actualizadas de la tecnología, no existen documentación que permita su soporte y actualización.

No se adaptan a los procedimientos actuales del Centro de Idiomas. No tienen en cuenta datos y procesos que se llevan durante la gestión de acceso y servicios en los CASIE.

No presentan una arquitectura que les permita la adaptabilidad y reusabilidad necesarios para agregar nuevas funciones y adaptarlos a nuevos requerimientos.

Por estos motivos, aunque su estudio contribuyó a determinar el estado actual de la automatización del negocio, es necesario realizar un estudio de las tecnologías y herramientas actuales que permitan el desarrollo de un nuevo sistema como posible solución al problema planteado.

### **1.3 Tecnologías y herramientas de desarrollo**

#### ***1.3.1 Elección de la metodología de desarrollo de software***

Una metodología de desarrollo desde el punto de vista del mundo de la informática, es un conjunto de técnicas y métodos que de manera integrada, permiten abordar cada una de las actividades del ciclo de vida de un proyecto de software. La metodología es un proceso completo y detallado que se utiliza para: (14).

- Facilitar la tarea de planificación.
- Facilitar la tarea del control y seguimiento de un proyecto.
- Mejorar la relación costo/beneficio.
- Optimizar el uso de recursos disponibles.
- Facilitar la evaluación de resultados y cumplimiento de los objetivos.
- Facilitar la comunicación efectiva entre usuarios y desarrolladores.

Actualmente, existen varias propuestas de metodologías que en dependencia del tipo de proyecto pueden contribuir a su éxito, por tanto, los desarrolladores deben evaluar efectivamente las características de cada empresa y cada metodología. En este punto, existen dos grandes corrientes, las metodologías ágiles y las metodologías tradicionales o pesadas.

No se puede decir que existe una metodología única para garantizar el éxito de cualquier proyecto de desarrollo de software, debido a que estas deben ajustarse al tipo de proyecto que se desarrolle.

De las metodologías tradicionales una de las más reconocidas y utilizadas es *RationalUnifiedProcess* (RUP).

### **RationalUnifiedProcess (RUP)**

RUP es un proceso de ingeniería planteado por Kruchten en 1996 cuyo objetivo es producir software de alta calidad, lo que significa desarrollar un software que cumpla todos los requerimientos planteados por el usuario dentro de una planificación y presupuesto establecido, cubriendo todo el ciclo de vida del producto (15). Es una metodología lo suficientemente robusta como para desarrollar grandes proyectos y lo suficientemente flexible como para adaptarse a pequeños proyectos (16).

Toma en cuenta las mejores prácticas de desarrollo, como la de una arquitectura basada en componentes, el manejo de requerimientos, el proceso iterativo, la modelación visual del software y el control de cambios. Como características fundamentales se definen (17):

- Dirigido por casos de uso: El proceso de desarrollo se dirige a partir de los casos de uso identificados durante la definición del negocio. Al dirigirse por caso de uso RUP permite un desarrollo organizado, separando las funcionalidades en paquetes y contribuye a una documentación del proyecto.
- Iterativo e incremental: El software se construye por etapas, iterando según la relevancia de los requisitos, teniéndose un desarrollo incremental con entregas probadas al cliente.
- Centrado en arquitectura: La línea base de la arquitectura se define al inicio de proyecto y sobre ella se desarrolla el sistema. Se obtiene un producto robusto con una arquitectura bien documentada.

RUP tiene 4 fases definidas, en las que intervienen 7 flujos de trabajo o disciplinas que agrupan en forma lógica las actividades a realizar. De estos flujos 4 son principales y 3 de apoyo.

Las fases son (18):

1. Comienzo o inicio: Se describe el negocio y se delimita el proyecto, describiendo el alcance del mismo con la definición de los casos de uso del negocio.
2. Elaboración: Se define la arquitectura del sistema y se obtiene un prototipo que responde a los requerimientos identificados.
3. Construcción: Se obtiene un producto listo para la utilización que está documentado y tiene manual de usuario. Se obtienen tantas versiones de entrega como determinan las pruebas.
4. Transición: Con la liberación, ya el producto está listo para su instalación en condiciones reales. Puede implicar la reparación de errores como parte del proceso de soporte.



Una descripción gráfica del proceso de desarrollo utilizando RUP puede documentar el esfuerzo empleado por el equipo en cada fase del proyecto (17):

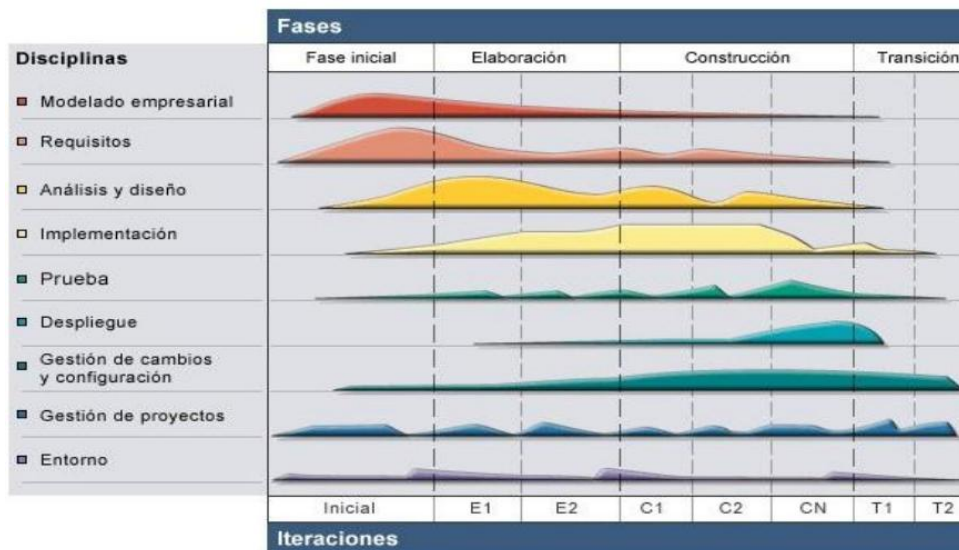


Figura 1. Fases de la metodología RUP

RUP es una metodología de desarrollo completa, adaptable y documentada que permite comprender y extender un proyecto de software. Su proceso iterativo permite desarrollar un producto por fases, centrándose en la arquitectura candidata, lo que permite asegurar que se utilice siempre el principio de diseño escogido, maximizando las posibilidades de éxito en todas las etapas del desarrollo.

Aunque es una metodología muy completa, RUP no es adecuada para proyectos con equipos de desarrollo pequeños, debido a que el esfuerzo para realizar todas sus fases es alto; además, requiere la especialización de los desarrolladores, inversión en tiempo, y que el negocio a automatizar tenga sus requisitos y características bien definidas. Ante esta situación, es necesario escoger una metodología que, sin descuidar la documentación y las fases establecidas para el proceso, asegure un desarrollo ágil y adaptable a proyectos pequeños. Por tanto, se analiza OpenUP, la versión ágil de RUP.

## OpenUP

Mantiene las mismas características de RUP en cuanto a desarrollo iterativo, casos de uso y escenarios de conducción de desarrollo, gestión de riesgos y el enfoque centrado en la arquitectura, basándose en un desarrollo ágil al eliminar la parte poco extensible y rígida de la metodología tradicional. Por tanto, es apropiado para proyectos pequeños o de bajos recursos, aplicable a un conjunto amplio de plataformas de desarrollo, su punto de vista pragmático le permite centrarse en una naturaleza colaborativa dentro del

equipo de desarrollo. Una de sus principales características es su alto grado de adaptabilidad a las necesidades de un proyecto en particular (18).

Como metodología de desarrollo es conducida por el principio de colaboración para alinear intereses y para compartir su comprensión. Es el proceso unificado que aplica acercamientos iterativos e incrementales dentro de un ciclo vital estructurado. Las capas de OpenUP pueden observarse a continuación.

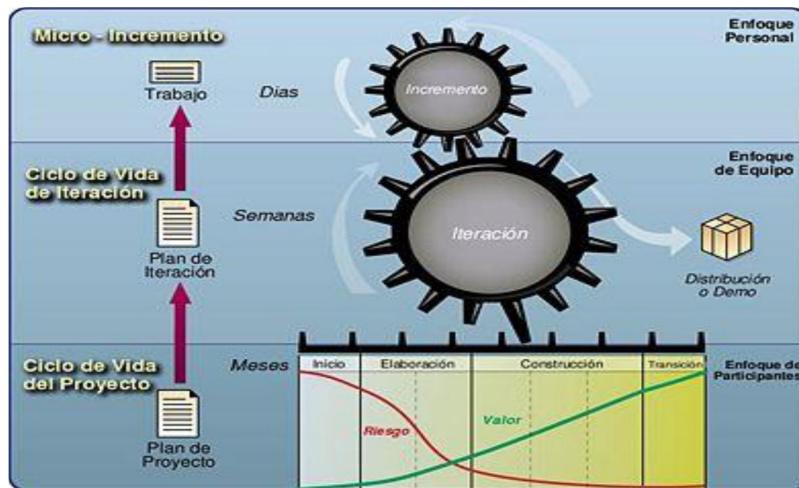


Figura 2. Capas de OpenUP

El ciclo de vida del proyecto provee a los interesados un mecanismo de supervisión y dirección para controlar los fundamentos del proyecto, su ámbito, la exposición a los riesgos, el aumento de valor y otros aspectos. OpenUp estructura el ciclo de vida de un proyecto en cuatro fases: inicio, elaboración, construcción y transición.

La elección de OpenUP sobre otras metodologías ágiles se basa en su similitud con RUP permitiendo elegir los artefactos a generar, su adaptación a proyectos pequeños, el desarrollo basado en casos de uso y la determinación de la Universidad de las Ciencias Informáticas en su utilización como metodología de software.

### 1.3.2 Lenguaje de modelado

Un lenguaje de modelado es un recurso para especificar, visualizar, construir y documentar los artefactos que se generan durante la construcción de un sistema de software. Aunque existen muchos lenguajes con estos fines (Ecore, MOF, etc.) el más estandarizado es el Lenguaje de Modelado Unificado (UML), actualmente en su versión 2 (19).

## Leguaje de Modelado Unificado (UML 2.0)

El Lenguaje de Modelado Unificado (en inglés *UnifiedModelLanguage*), incluye conceptos semánticos, notación y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. UML es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño. Existen diferencias importantes entre un método y un lenguaje de modelado. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código, así como generadores de informes. La especificación de UML no define un proceso estándar, pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. Es una consolidación de muchas de las notaciones y conceptos más usados en la modelación orientada a objetos. Es importante (19) recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, no es un proceso, sino un lenguaje que permite la modelación de sistemas, con tecnología orientada a objetos. Desde el año 1995, UML es un estándar aprobado por la ISO como ISO/IEC 19501:2005 *Informationtechnology*; está respaldado por *el Object Management Group (OMG)*.

Un modelo UML está compuesto por tres clases de bloques de construcción:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- Relaciones: relaciona los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones.

Permite la modelación del ciclo completo de desarrollo de software y contiene: Los Diagramas de casos de uso, clases, objetos, secuencia, colaboración, estado, actividades, despliegue y componentes.

Los principales beneficios de UML son (20):

- Mejores tiempos totales de desarrollo (de la mitad del desarrollo o más).
- Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos.

La elección de UML como lenguaje de modelado responde a la necesidad de representar los artefactos generados durante el proceso de desarrollo utilizando un lenguaje común, fácil y completo, de forma que se asegure la comprensión de dichos artefactos y la mejora continua del sistema.

### **1.3.3 Lenguaje de programación**

Un lenguaje de programación es una representación artificial diseñada para expresar procesos que pueden interpretarse y ejecutarse por computadoras y otras máquinas. Pueden utilizarse para controlar el comportamiento físico y lógico de una máquina, expresar algoritmos y modos de comunicación. Lo componen símbolos, reglas y semánticas que determinan su estructura. La selección de cada lenguaje se realiza de acuerdo a las características y al tipo de software a desarrollar (21).

El tipo de sistema está determinado por el ambiente, los componentes que lo componen y las necesidades de los clientes. Siguiendo estos detalles, para el problema planteado en esta investigación, la solución de software será un sistema web. Las tendencias actuales del desarrollo de aplicaciones web implican como lenguajes candidatos JavaScript, HTML, CSS y PHP.

#### **JavaScript**

Es un lenguaje de programación interpretado, ligero y orientado a objetos, se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. A pesar de su nombre, no guarda ninguna relación directa con el lenguaje de programación Java, tienen semánticas y propósitos diferentes. Los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web (22). La versión actual de JavaScript es la 2, elegida para el desarrollo y soportada por todos los navegadores web.

#### **PHP**

*HypertextPreprocessor*(PHP) o Procesador Hipertexto en español, es un lenguaje etiquetado diseñado para incrementar el dinamismo con las páginas web, aprovechando los beneficios de la red. Es un lenguaje muy difundido flexible, de alto nivel y embebido dentro de las páginas HTML. Se ejecuta del lado del servidor y permite insertar contenidos dinámicamente. Con la versión 5.4 se solucionaron algunos aspectos que se echaron en falta en versiones anteriores. El principal objetivo de PHP 5 ha sido mejorar los mecanismos de programación orientada a objetos. Un paso necesario para lograr que PHP sea un lenguaje apto para todo tipo de aplicaciones y entornos. Es el lenguaje más utilizado en la web para la

realización de páginas avanzadas para todo tipo de entornos (23). En la aplicación se hará uso de la versión 5.4.2.

### **CSS 3**

Las Hojas de Estilos en Cascada (CSS, por sus siglas en inglés) es un lenguaje creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas Web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien formateados y con significado completo (también llamados documentos semánticos). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes (24). Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, además de otras funcionalidades.

### **HTML 5**

Siglas de *Hyper Text Markup Language* o Lenguaje de Marcas de Hipertexto en español, es el lenguaje de marcado más utilizado para la construcción de páginas web. Es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la síntesis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica como desplegar el contenido del documento, incluyendo texto imágenes y otros medios soportados. Su versión 5 brinda más facilidades que sus antecesores, lo que permite el desarrollo de la aplicación con una mayor vistosidad, calidad e integración (22).

Los lenguajes a utilizar determinan las tecnologías para apoyar el proceso de desarrollo. En este caso, es imprescindible elegir un marco de trabajo que propicie velocidad, buenas prácticas y seguridad.

Estos lenguajes de programación marcan la referencia en el desarrollo de las aplicaciones web, son los más utilizados dentro de las infraestructuras de internet. En todos los casos existen una extensa documentación, constantes actualizaciones y máxima compatibilidad con todas las arquitecturas y herramientas orientadas a la web.

#### **1.3.4 Elección del framework de desarrollo**

Un framework de desarrollo o marco de trabajo en español, es un conjunto de herramientas enfocadas en la obtención de una aplicación de software. Es una estructura conceptual que simplifica el desarrollo mediante la automatización de tareas comunes a través de patrones de diseño. Evita la re-implementación de funcionalidades que resultan ser frecuentes en los sistemas y que tradicionalmente incluyen

administración de usuarios, persistencia de datos, motores de plantillas, entre otras; facilita la programación de aplicaciones, puesto que encapsula operaciones complejas en instrucciones sencillas y además, proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener (25).

Entre los framework más utilizados en el desarrollo de aplicaciones web pueden citarse (26):

Codeigniter: Es un marco de trabajo PHP que parte de un núcleo muy pequeño.

Cakephp: Es un marco de trabajo PHP para el desarrollo rápido de aplicaciones que presenta una arquitectura extensible.

Symfony: Marco de trabajo PHP que provee componentes y herramientas suficientes para escribir aplicaciones robustas.

PradoSoft: Es un marco de trabajo basado en componentes y manejado por eventos.

Aunque para el desarrollo de aplicaciones basadas en PHP existen diversos marcos, la opción de Symfony es la elegida por:

Symfony es un framework PHP de alto rendimiento diseñado a partir de varios principios y propuestas de varios subsistemas que le permiten desarrollar aplicaciones robustas para web. Su arquitectura es totalmente desacoplada, lo que le permite ajustar todas sus partes, eligiendo aquellas que responden o no a los proyectos. Utiliza el patrón Modelo-Vista-Controlador y de esa misma forma organiza el código; permite escribir aplicaciones de forma elegante y directa, además de utilizar rutinas y componentes para la generación automática de vistas, seguridad, integración y acceso a datos. (27). La versión 2 se publicó en junio de 2013 e incorpora varias mejoras en la configuración, el uso de funciones, la integración y la compatibilidad por lo que la versión a utilizar es 2.7.

Para el uso de JavaScript y trabajo con las vistas se utiliza el marco de trabajo Ext JS en su versión 4.2.1. Ext JS es una librería de JavaScript ligera, de alto rendimiento para el desarrollo rápido de aplicaciones web cross-browser. Presenta una interfaz de usuario personalizable “*widgets*”, bien diseñada y contiene un modelo de componentes extensibles. Tiene disponibles licencias comerciales y de código abierto. Es compatible con la mayoría de los navegadores (28).

### **1.3.5 Sistema gestor de Base de Datos**

Un sistema de gestión de base de datos (SGBD) es un programa que permite almacenar y acceder a datos que guarda de forma rápida y estructurada. Incluyen un lenguaje de definición de datos, un lenguaje

de manipulación y un lenguaje de consulta. Debe darle al usuario la posibilidad de manipular los datos garantizando la integridad de la información (29).

Entre los sistemas de base de datos relacionales los más utilizados son PostgreSQL, MySQL, Sybase, Oracle, Microsoft SQL Server entre otros. La mayoría de las herramientas de software libre para la publicación de contenidos web (*webservers*) incluyen MySQL en su versión 5.6 como motor de base de datos, característica que lo hace elegible, además de su integración total al framework seleccionado.

Es un sistema de gestión de base de datos objeto-relacional caracterizado por utilizar un modelo cliente-servidor. Emplea multihilos para asegurar la estabilidad del sistema de modo que un fallo en uno de los hilos no afecta el resto de ellos, de esta forma el sistema continuará funcionando sin dificultades. Distribuido bajo la licencia GNU GPL y propiedad de Oracle, el código fuente disponible es uno de los sistemas de gestión de base de datos más reconocidos del mercado. Se encuentra incluido en muchos de los servidores web de código abierto y entre sus principales características se encuentran (30):

- Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas.
- Vistas.
- Integridad transaccional.
- Acceso concurrente multiversión (no se bloquean las tablas, ni siquiera las filas, cuando un proceso escribe).
- Herencia de tablas.

En los sistemas comerciales actuales, los tipos posibles incluyen numéricos de punto flotante, enteros, cadenas de caracteres, cantidades monetarias y fechas.

### **1.3.6 Herramientas de apoyo para el desarrollo de software**

Dentro de las herramientas necesarias para la realización del sistema se encuentran el Entorno de Desarrollo Integrado, herramienta CASE y Servidor Web.

Un Entorno de Desarrollo Integrado, (IDE de sus siglas en inglés *IntegratedDevelopmentEnvironment*) consiste básicamente en un software cuyo principal objetivo es el desarrollo de otro software. Es un entorno de programación que ha sido empaquetado como un programa de aplicación. Puede ser exclusivo para un lenguaje de programación o bien, para varios. Suele consistir en un editor de código (con facilidades como resaltado de sintaxis, completamiento de código y navegación entre clases), un compilador y herramientas de automatización de la compilación, un depurador y en algunos casos un constructor de interfaz gráfica (21).

## IDE NeatBeans

Neatbeans es un entorno de desarrollo integrado libre, extensible para el desarrollo sobre muchos lenguajes, aunque se realizó fundamentalmente para el lenguaje Java<sup>1</sup>. Es un producto de código abierto desarrollado por la compañía SunMicroSystem en el año 2000. Se basa en una filosofía modular, lo que permite el desarrollo de múltiples proyectos con el uso de varias tecnologías (30). Permite a los programadores escribir, compilar, depurar y ejecutar programas. Provee la implementación del patrón de arquitectura Modelo-Vista-Controlador, además de estar enfocado para el desarrollo del lenguaje dinámico de programación PHP. Es fácil de instalar y configurar en la mayoría de las plataformas. Entre sus principales ventajas se encuentran:

- Administración de interfaces de usuario.
- Integración a múltiples frameworks o marcos de trabajo en español.
- Administración de almacenamiento.
- Gran cantidad de módulos y extensiones para múltiples lenguajes y tecnologías.
- Fuerte comunidad de respaldo.

Es utilizado en la aplicación en su versión 8 debido a que brinda la posibilidad de facilitar y automatizar una gran cantidad de funcionalidades que resultan comunes en las aplicaciones web, permite el auto completado del código, se integra con el framework seleccionado, asegura el versionado del código y la depuración de aplicaciones.

## Herramienta Case

Las herramientas CASE, Ingeniería de Software Asistida por Computadoras (por sus siglas en inglés *ComputerAided Software Engineering*) son un conjunto de programas y ayudas que propician el desarrollo de programas informáticos a los ingenieros de software y desarrolladores. Las mismas permiten automatizar diferentes tareas que se realizan durante el ciclo de vida de desarrollo del software, tales como: la documentación, la generación de código y diseños, las pruebas de errores y la gestión del proyecto. Permite la estandarización de la documentación y la reutilización del software (19). Entre las múltiples propuestas, Visual Paradigm es una de las mejores herramientas, además de tener un uso estandarizado dentro de la Universidad de Ciencias Informáticas.

## Visual Paradigm for UML 8.0

---

<sup>1</sup> Java: Lenguaje de programación orientado a objetos desarrollado por SunMicroSystem en los años 90.



Es una herramienta considerada muy completa, fácil de usar, con soporte multiplataforma y de probada utilidad para el analista, proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Posee un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, análisis y diseño, hasta la generación del código fuente y la documentación. Esta herramienta se utilizará para visualizar y diseñar los elementos de nuestra aplicación debido a que es multiplataforma. Se puede intercambiar diagramas UML y modelos con otras herramientas. Permite la generación de la base de datos a partir de un modelo de diseño. Posee facilidades para el diseño de los diagramas necesarios y su documentación, soporta las últimas versiones de UML y la Notación y Modelado de Procesos de Negocios, a través de la utilización de un enfoque Orientado a Objetos con abundante documentación (31).

### **Servidor web**

Un servidor web es un programa que implementa el protocolo HTTP y se encarga de mantenerse a la espera de peticiones llevadas a cabo por un cliente que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos (32).

### **Servidor Web Apache 2.4**

Apache es la plataforma de servidores web de código fuente abierto más utilizada en el mercado. Su configuración, robustez y estabilidad hacen que más del 64% de toda web reiteren su confianza en este programa. Es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos. Apache es una de las plataformas de servidores Web más destacadas dentro de la gran cantidad de servidores web que existen en el planeta, de código fuente abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Entre sus características principales se encuentran (32):

- Tecnología gratuita de código fuente abierta.
- Personalizable, la arquitectura modular de Apache permite construir un servidor hecho a la medida y posibilita la implementación de los últimos y nuevos protocolos.
- Servidor altamente configurable de diseño modular. Permite aumentar fácilmente su capacidad e instalar cualquier módulo para cumplir una función específica.

- Tiene la infraestructura necesaria para servir distintos protocolos.
- Rapidez y estabilidad en sistemas que no son tipo Unix, tales como BeOS, OS/2 y Windows.
- Nueva interfaz de programación (API) para los módulos, muchos de los problemas de ordenación y prioridad de módulos de la versión 1.3 desaparecieron.
- Los módulos de Apache pueden escribirse para que se comporten como filtros que actúan sobre el flujo de contenidos tal y como salen del servidor, o tal y como son recibidos por el servidor.

#### **1.4 Conclusiones parciales**

A partir de los temas abordados en el capítulo se puede concluir que: La identificación y estudio de los principales conceptos relacionados con la gestión de Información en los Centros de Idiomas permitió conocer los flujos de procesos involucrados en el campo de acción, lo que contribuyó a formular las bases teóricas de la investigación. La selección de las tecnologías web basadas en Symfony 2 como marco de trabajo y ExtJS para la presentación al usuario, permitirán desarrollar un sistema acorde a las necesidades de automatización presentes en el Centro de Idiomas de la Universidad de las Ciencias Informáticas. El estudio de los sistemas analizados, permitió algunos de los requisitos relacionados con la problemática y ayudaron a justificar la realización de un nuevo sistema de gestión.

## Capítulo 2 Análisis y diseño del sistema

### Introducción

En este capítulo se definirá el modelo de negocio que responde al proceso de control de acceso y actividades de los CASIE, a través de las reglas, actores y casos de uso relacionados, lo que servirá para establecer una propuesta de solución. Para detallar la propuesta, se expondrán los requisitos definidos y se abordará el diseño de la herramienta a partir de sus paquetes, arquitectura, modelo de datos y diagrama de despliegue.

### 2.1 Modelo de Negocio

#### 2.1.1 Descripción del negocio

EL negocio tiene lugar en los CASIE que pertenecen al Centro de Idiomas de la Universidad de las Ciencias Informáticas. No incluye ninguna entidad externa y todas las tareas están relacionadas con dos procesos fundamentales: el control de acceso a las salas de los CASIE y los servicios que se brindan dentro de dichas áreas. Una descripción general puede apreciarse a partir de las reglas establecidas, los actores involucrados y la descripción de los casos de uso.

#### 2.1.2 Reglas del negocio a considerar

Las principales reglas a considerar dentro del negocio son las siguientes:

1. Todo acceso a salones de los CASIE debe ser registrado para su futuro análisis.
2. El registro es realizado exclusivamente por el Técnico General, en su ausencia, el Técnico Informático debe realizar el registro de acceso.
3. Las actividades en los salones se realizan según la disponibilidad de estos, y teniendo en cuenta la planificación docente establecida por el Centro de Idiomas para cada CASIE.
4. Los técnicos deben reportar cualquier incidencia que se establezca con el uso de recursos y materiales.

#### 2.1.3 Actores del negocio

El actor del negocio es el técnico general que realiza las dos actividades del negocio de controlar acceso y gestionar los préstamos, como puede verse en el siguiente diagrama de casos de uso del negocio.

#### 2.1.4 Diagrama de casos de uso del negocio

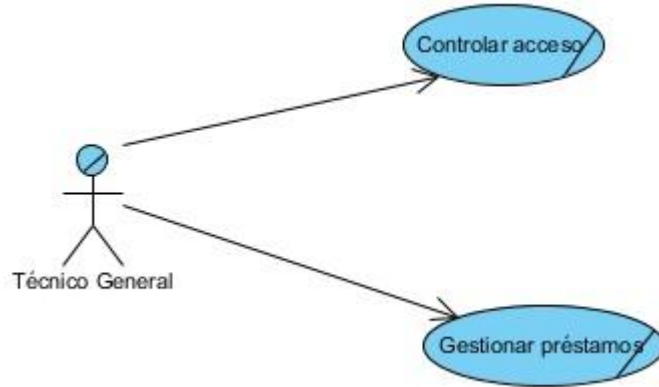


Figura 3. Diagrama de casos de uso del negocio

### 2.1.5 Descripción de casos de uso del negocio

La descripción de cada Caso de Uso del Negocio puede observarse mediante los diagramas de actividades. Las actividades señaladas en gris son posibles tareas a automatizar.

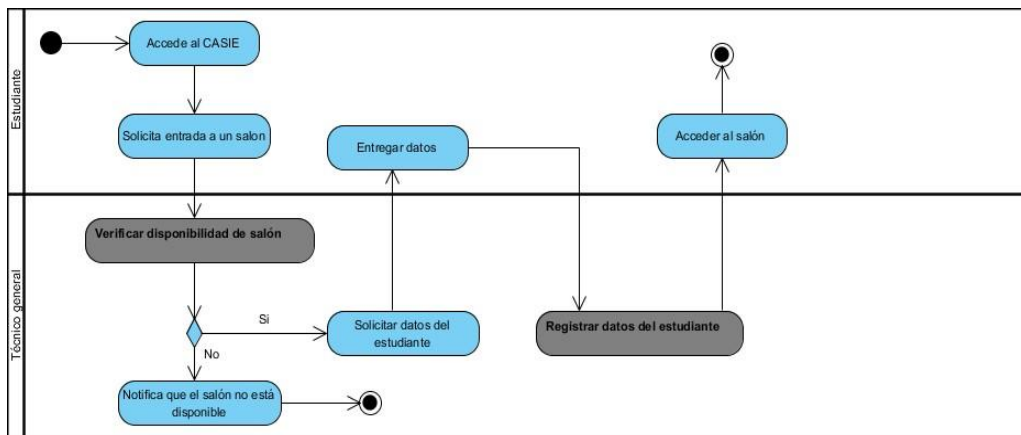


Tabla 1. Diagrama de actividades caso de uso Controlar Acceso

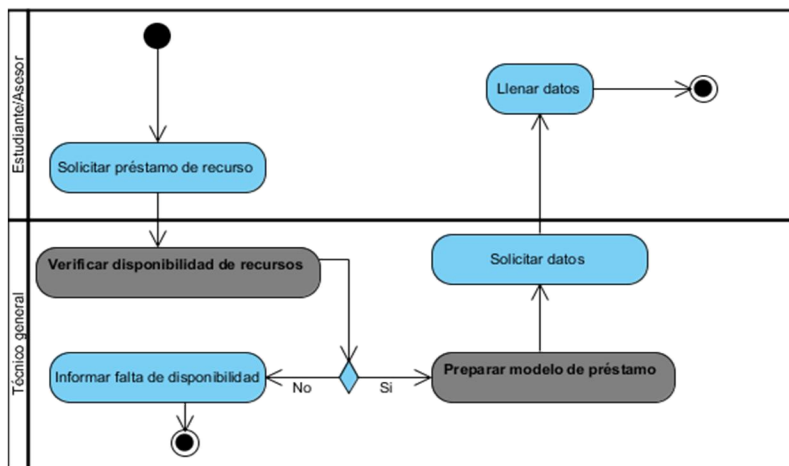


Tabla 2. Diagrama de actividades caso de uso Gestionar Préstamos

### **2.1.6 Propuesta de solución.**

A partir del negocio detallado se propone la realización de un sistema para el control de acceso y servicios prestados por el Centro de Idiomas de la Universidad de las Ciencias Informáticas. El sistema estará orientado a los CASIE, donde los técnicos podrán realizar la gestión de préstamos y control de acceso, además de permitir la planificación de actividades y las reservaciones de espacios. El sistema debe permitir extraer información útil para la comprensión y análisis de los procesos, además de proveer seguridad. Las funcionalidades y características que debe tener la aplicación pueden observarse mediante sus requisitos de software.

## **2.2 Requisitos de software**

Los requisitos de software son la clave para el desarrollo de aplicaciones informáticas, son precisamente, lo que el software debe hacer o cumplir. Los requisitos pueden ser funcionales o no funcionales, y su manejo constituye una de las disciplinas dentro del ámbito informático (15). Seguidamente se definen los requisitos funcionales del sistema, así como sus actores y relaciones a partir de los casos de uso; además de los requisitos no funcionales.

### **2.2.1 Requisitos funcionales**

Los requisitos funcionales son las capacidades de resolución que el sistema debe tener. Los requisitos funcionales son los siguientes:

RF 1. Autenticar Usuario

RF 2. Gestionar Usuarios

2.1. Listar usuario

2.2. Adicionar usuario

2.3. Modificar usuario

2.4. Eliminar usuario

RF 3. Gestionar Centros

3.1. Listar centros

3.2. Adicionar centros

3.3. Eliminar centros

3.4. Modificar centros

RF 4. Gestionar Servicios

4.1. Listar servicios

4.2. Adicionar servicio

4.3. Modificar servicio

4.4. Eliminar servicio

RF 5. Asociar servicios a centros

RF 6. Gestionar Recursos

6.1. Listar recursos

6.2. Adicionar recurso

6.3. Eliminar recurso

6.4. Modificar recurso

RF 7. Gestionar Idiomas

7.1. Listar idiomas

7.2. Adicionar idioma

7.3. Modificar idioma

7.4. Eliminar idioma

RF 8. Gestionar Actividades

8.1. Listar actividades

8.2. Adicionar actividad

8.3. Modificar actividad

8.4. Eliminar actividad

RF 9. Gestionar Préstamos

9.1. Listar préstamos

9.2. Abrir préstamo

9.3. Cerrar préstamo

RF 10. Gestionar Asesores

10.1. Listar asesores

10.2. Adicionar asesor

10.3. Modificar asesor

10.4. Eliminar asesor

RF 11. Controlar Acceso

11.1. Registrar entrada

11.2. Registrar salida

RF 12. Reservar Horario.

RF 13. Cancelar Reservación

RF 14. Gestionar Reservación

- 14.1. Aprobar reservación
- 14.2. Cerrar reservación
- RF 15. Imprimir Reportes
  - 15.1. Imprimir reporte de préstamos
  - 15.2. Imprimir reporte de acceso
  - 15.3. Imprimir reporte de reservación
- RF 16. Visualizar Reportes Gráficos.
  - 16.1. Visualizar gráfico de acceso por actividad
  - 16.2. Visualizar gráfico de acceso por servicio
  - 16.3. Visualizar gráfico de uso de servicios
  - 16.4. Visualizar gráfico de reservaciones de servicios
  - 16.5. Visualizar gráfico de préstamos por centro

### **2.2.2 Requisitos no funcionales**

**Requisitos de usabilidad:** El sistema debe ser simple, con un menú superior accesible desde todas las páginas. Debe garantizarse una curva de aprendizaje alta, de modo que los técnicos, asesores y personal del Centro de Idiomas se familiaricen rápidamente con la interfaz.

**Requisitos de seguridad:** El sistema establecerá una política de acceso basada en roles y usuarios, debe existir un rol para los técnicos, uno para los asesores, para el análisis de datos y administración. Debe garantizarse la integridad, disponibilidad y confidencialidad de la siguiente forma:

- **Integridad:** La información generada debe ser consistente y protegida contra alteraciones de cualquier tipo. Durante la manipulación de informaciones de los procesos, éstas no pueden ser alteradas por otros usuarios.
- **Disponibilidad:** El sistema deberá estar disponible durante todo el tiempo laboral, y permitir el acceso desde todos los CASIE y desde la Dirección del Centro de Idiomas.
- **Confidencialidad:** El acceso se realizará mediante usuario UCI.

**Requisitos de interfaces externas:** La interfaz debe tener colores serios, sobre las gamas frías. Los botones deben cumplir patrones estándar de formas e íconos. Las interfaces deben ser ajustables a los diferentes tipos de resoluciones de un ordenador estándar sin perjudicar la visualización de la información durante el reajuste de los navegadores.

**Requisitos de rendimiento:** La aplicación debe ejecutarse utilizando eficientemente los recursos de software y hardware en el lado del servidor, y además debe asegurarse que los tiempos de respuesta a las diferentes peticiones de los usuarios sea por debajo de los 5 segundos.

Requisitos de Software para el cliente:

- Navegador web (Opera 10+, Mozilla Firefox20+, Internet Explorer 9+, Safari 3+).

Requisitos de Software para el servidor

- Servidor Web Apache 2.2 o superior.
- Servidor de Base de Datos MySQL 5 o superior.
- PHP versión 5.4.2 o superior

Requisitos de Hardware del servidor:

- Procesador Pentium/AMD, 2.4 GHz o superior.
- Memoria RAM: 2 Gb.
- Disco duro: 40 Gb o superior.
- Tarjeta de Red o módem.

### 2.2.3 Actores del sistema

Los actores del sistema se relacionan en la siguiente tabla.

Actor del sistema	Descripción
<b>Usuario</b>	Constituye todo usuario que accede al sistema.
<b>Administrador</b>	Usuario con acceso a todas las interfaces del sistema. Es el encargado de realizar las configuraciones iniciales, además de gestionar los usuarios que acceden al sistema.
<b>Técnico general</b>	Usuario principal del sistema, realiza el control de acceso, gestión de recursos, préstamos y reservaciones, además de emitir reportes.
<b>Asesor</b>	Usuario que realiza reservaciones de salas en el sistema.
<b>Jefe de centro de idioma</b>	Usuario que visualiza los reportes generados por el sistema

Tabla 3. Actores del sistema

### 2.2.4 Diagrama de casos de uso del sistema



El diagrama de casos de uso del sistema, documenta el comportamiento de un software desde el punto de vista del usuario. Por tanto los casos de uso determinan los requisitos funcionales del sistema, los cuales representan las funcionalidades que un sistema puede ejecutar (18).

A continuación se presenta el diagrama de casos de uso del sistema:

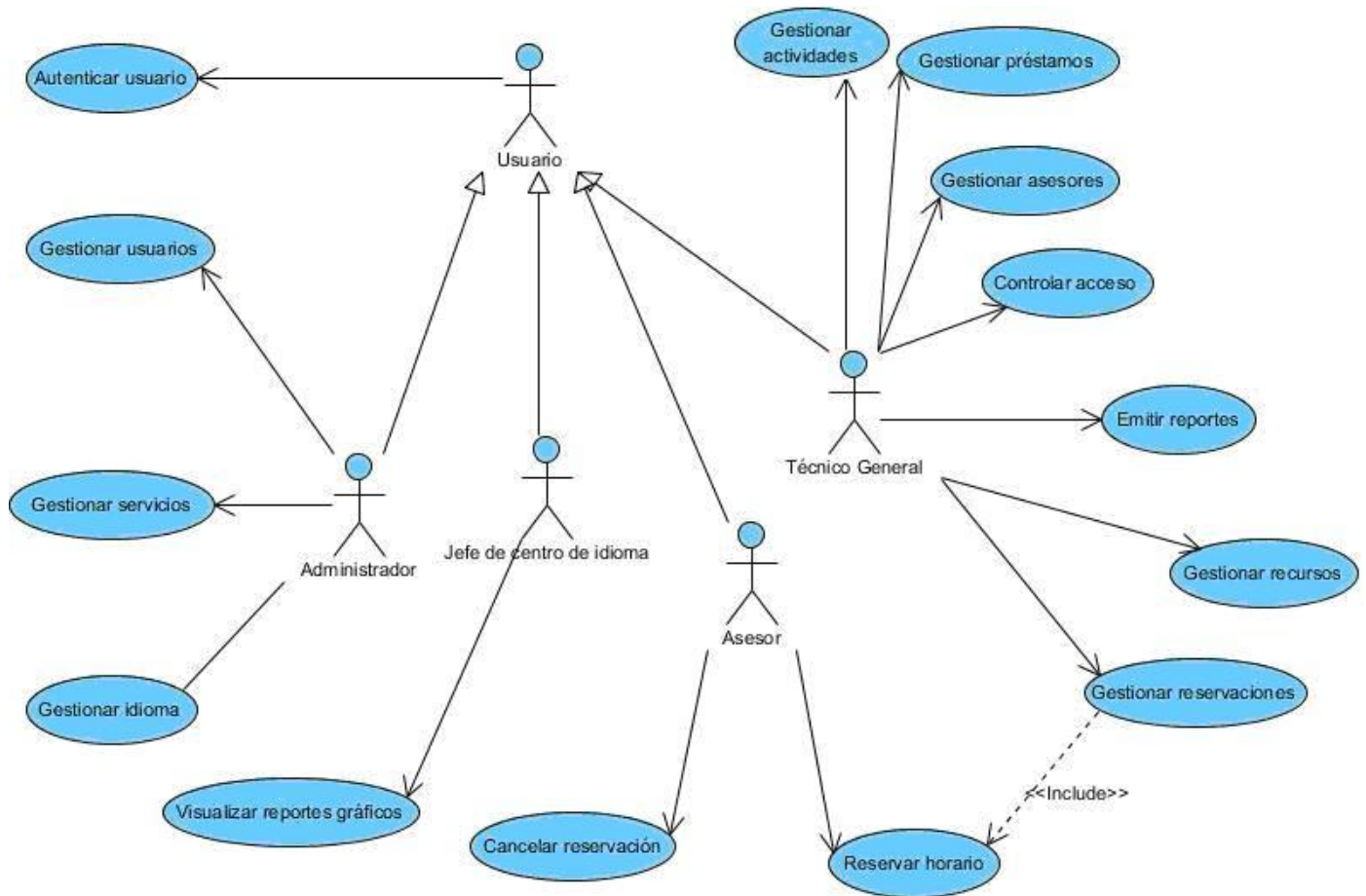


Figura 4. Diagrama de casos de uso del sistema

### 2.2.5 Especificaciones de caso de uso

Se especifican los casos de uso más relevantes del sistema. Los restantes casos de uso pueden consultarse dentro de los anexos de la investigación.

<b>Caso de uso</b>	Autenticar usuario
<b>Objetivo</b>	Dejar autenticado un usuario en el sistema
<b>Actores</b>	Usuario (Administrador, Jefe de centro de idiomas, Asesor, Técnico general)
<b>Resumen</b>	Ocurre cuando el usuario accede a administración desde el sitio del

	centro de idiomas y se autentica para realizar actividades de gestión.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Crítico [Crítico, Secundario, Auxiliar, Opcional ]	
<b>Precondiciones</b>	Ninguna	
<b>Postcondiciones</b>	El usuario ha quedado autenticado en el sistema y ha iniciado la sesión.	
<b>Flujo de eventos</b>		
<b>Flujo básico Autenticar usuario</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Accede a administración desde el menú principal del sitio del Centro de Idiomas	
2.		Muestra el formulario de autenticación con los campos a llenar de usuario y contraseña
3.	Introduce los campos de usuario y contraseña	
4.		El sistema valida que el usuario y la contraseña sean correctos. En caso afirmativo se autentica el usuario e inicia la sesión. En caso de que el usuario o la contraseña sean incorrectas se Ejecuta en Evento alternativo No. 1
<b>Flujos alternos</b>		
<b>1. Usuario o contraseña incorrectos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		El sistema muestra un mensaje de: "Su usuario no aparece con permisos o su contraseña no es correcta. Verifique sus datos de acceso"
2.	Realiza nuevamente la tarea dos del flujo Básico	
<b>Requisitos funcionales</b>	<b>no</b>	Requisitos de seguridad
<b>Asuntos</b>		

<b>pendientes</b>	
-------------------	--

Tabla 4. Especificación Caso de uso Autenticar usuario

<b>Caso de uso</b>	Gestionar préstamos	
<b>Objetivo</b>	Visualizar, adicionar o cerrar préstamos realizados sobre recursos del centro de idiomas.	
<b>Actores</b>	Técnico general: (Inicia) Visualiza, Adiciona, y Cierra los préstamos en el sistema	
<b>Resumen</b>	El técnico general adiciona, visualiza o cierre los préstamos solicitados a su área.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El usuario se ha autenticado. El usuario tiene el rol de técnico general.	
<b>Postcondiciones</b>	Se gestiona uno o más préstamos de recursos.	
<b>Flujo de eventos</b>		
<b>Flujo básico Gestionar préstamo</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Accede a la funcionalidad préstamos de las operaciones del árbol de funcionalidades	
2.		Muestra una lista de los préstamos realizados que estarán ordenados descendientemente según la fecha. En caso de no existir préstamos ocurre el evento No 1 del flujo alterno.  El usuario podrá <ul style="list-style-type: none"> <li>- Adicionar un nuevo préstamo (Ver sección Adicionar préstamo)</li> <li>- Cerrar préstamo (Ver sección cerrar préstamo)</li> </ul>
<b>Flujos alternos</b>		
<b>No 1 No existen préstamos</b>		

	<b>Actor</b>	<b>Sistema</b>
1.		Muestra la lista préstamos vacía con el mensaje: "No existen préstamos definidos".
<b>Sección 1: "Adicionar préstamo"</b>		
<b>Flujo básico Adicionar préstamo</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Escoge la opción adicionar préstamo	
2.		Muestra un formulario para el registro del préstamo. El formulario contiene los campos de: Recurso, usuario de la persona que solicita el préstamo, y fecha prevista de entrega.
3.	Introduce los datos en los campos del formulario	
4.		Valida los datos introducidos por el usuario. Si el usuario no existe se produce el flujo alterno No 2. Si el recurso está prestado ocurre el flujo alterno No.3. Si algunos de los datos están sin llenar se produce el flujo alterno No. 4.  Muestra al sistema el mensaje: "Una vez definidos los datos no podrá modificarlos ni eliminarlos. ¿Está seguro que los datos son correctos?"
5.	Presiona "Si"	
6.		Se registran los datos del préstamo en el sistema. Concluye el caso de uso.
<b>Flujos alternos</b>		
<b>Nº 2 Usuario no encontrado</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje de: "El usuario que solicita no se encuentra en el directorio. ¿Desea escoger otro

		usuario?”.
2.	El usuario presiona “Si”	
		Limpia el campo de usuario.
<b>Nº 3 Recurso prestado</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje de: “El recurso que está solicitando no se encuentra disponible. ¿Desea escoger otro recurso?”.
2.	El usuario presiona “Si”	
3.		Limpia el campo de recurso.
<b>Nº 4 Campos vacíos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Señala los campos en rojo con una marca diciendo: “Estos campos son obligatorios”
2.	Inicia la tarea 3 del flujo normal de eventos	
<b>Sección 2: “Cerrar préstamo”</b>		
<b>Flujo básico Cerrar préstamo</b>		
1.	Escoge la opción “Cerrar préstamo” en la barra de herramientas.	El usuario muestra el mensaje: “¿Está seguro que desea cerrar el préstamo?” Si no se ha seleccionado un préstamo pasa al flujo de eventos No 5.
2.	Escoge la opción “Si”	
3.		Registra la fecha de cierre en el préstamo. Actualiza la lista de préstamos. Concluye el caso de uso.
<b>Nº 5 No se selecciona un préstamo</b>		
		Muestra un mensaje de advertencia: “Debe seleccionar un préstamo”
<b>Relaciones</b>	<b>CU Incluidos</b>	

	<b>CU Extendidos</b>	
<b>Requisitos no funcionales</b>		
<b>Asuntos pendientes</b>		

Tabla 5. Especificación Caso de uso Gestionar préstamo

<b>Caso de uso</b>	Gestionar reservaciones	
<b>Objetivo</b>	Visualiza las reservaciones del asesor y permite realizar una nueva reservación	
<b>Actores</b>	Asesor: (Inicia) Visualiza, y adiciona las reservaciones en el sistema.	
<b>Resumen</b>	El asesor gestiona las reservaciones de las actividades en los CASIE	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Secundario	
<b>Precondiciones</b>	El usuario se ha autenticado. El usuario tiene el rol de asesor.	
<b>Postcondiciones</b>	Se gestiona una o más reservaciones.	
<b>Flujo de eventos</b>		
<b>Flujo básico Gestionar reservación</b>		
	<b>Actor</b>	<b>Sistema</b>
3.	Accede a la funcionalidad reservaciones de las operaciones del árbol de funcionalidades	
4.		Muestra una lista de las reservaciones realizadas por el asesor. En caso de no existir reservaciones ocurre el flujo alternativo No 1. El usuario podrá: <ul style="list-style-type: none"> <li>- Adicionar reservación (Ver sección Adicionar reservación)</li> <li>- Cancelar reservación (Ver caso de uso cancelar reservación)</li> </ul>
<b>Flujos alternos</b>		

<b>No 1 No existen reservaciones</b>		
	<b>Actor</b>	<b>Sistema</b>
2.		Muestra la lista préstamos vacía con el mensaje: “No existen reservaciones definidas”.
<b>Sección 1: “Adicionar reservación”</b>		
<b>Flujo básico Adicionar reservación</b>		
	<b>Actor</b>	<b>Sistema</b>
7.	Escoge la opción adicionar reservación	
8.		Muestra un formulario para el registro de la reservación. El formulario contiene los campos de: Centro, Actividad, Fecha y hora.
9.	Introduce los datos en los campos del formulario	
10.		<p>Valida los datos introducidos por el asesor. Si el horario y la fecha no están disponibles ejecuta el flujo alternativo No 2. Si algunos de los datos están sin llenar se produce el flujo alternativo No. 3.</p> <p>Muestra al sistema el mensaje: “Una vez definidos los datos no podrá modificarlos ni eliminarlos. ¿Está seguro que los datos son correctos?”</p>
11.	Presiona “Sí”	
12.		<p>Se registran los datos de la reservación en el sistema.</p> <p>Se envía un correo electrónico con los datos de la reservación al técnico general. Concluye el caso de uso.</p>
<b>Flujos alternos</b>		
<b>Nº 2 Horario ocupado</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje de: “El horario

		seleccionado está ocupado. ¿Desea escoger otro horario?”.
2.	El usuario presiona “Si”	
		Limpia los campos de Centro, Fecha y Hora.
<b>Nº 3 Campos vacíos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Señala los campos en rojo con una marca diciendo: “Estos campos son obligatorios”
2.	Inicia la tarea 3 del flujo normal de eventos	
<b>Relaciones</b>	<b>CU Incluidos</b>	<i>CU Reservar horario</i>
	<b>CU Extendidos</b>	
<b>Requisitos no funcionales</b>		
<b>Asuntos pendientes</b>		

Tabla 6. Especificación Caso de uso Gestionar reservaciones

## 2.3 Diseño de la solución

### 2.3.1 Arquitectura

La arquitectura a nivel de aplicación se basa en el patrón Modelo- Vista-Controlador. El principio más importante de la arquitectura MVC es la separación del código entres capas, dependiendo de su naturaleza. La lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador. La programación se puede simplificar si se utilizan otros patrones de diseño. De esta forma, las capas del modelo, la vista y el controlador se pueden subdividir en más capas (33).

En la solución la aplicación de este patrón de forma general se observa de la siguiente forma:



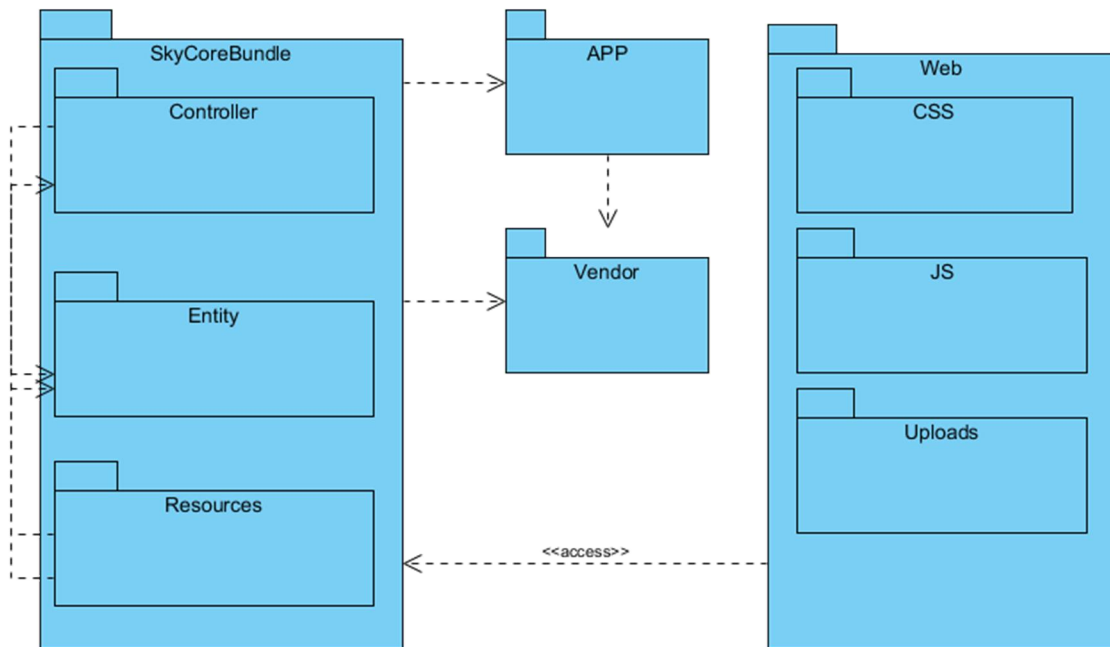


Figura 5. Diagrama de paquetes Sistema

El paquete SkyCoreBundle contiene los siguientes elementos: El paquete Controller que engloba las clases controladoras. Las entidades de persistencia de la base de datos que pertenecen al modelo, que se encuentran en el paquete Entity. Además, el paquete Resources contiene las vistas que se generan y son las encargadas de cargar los recursos presentes en el paquete Web, que tiene los recursos que pueden ser accedidos por el usuario. Los paquetes APP y Vendor contienen los archivos y configuraciones básicas del framework Symfony.

### Estilo arquitectónico

El estilo Cliente-Servidor es el que define una relación entre dos aplicaciones en las cuales una de ellas (cliente) envía peticiones a otra (servidor fuente de datos), para su procesamiento. Es un estilo para sistemas distribuidos que divide el sistema en una aplicación cliente, una aplicación servidora y una red que las conecte, en la cual la primera realiza peticiones y la segunda emite respuesta usando un amplio rango de protocolos y formatos de datos para comunicar información.

#### 2.3.2 Patrón de diseño

Un patrón de diseño es una buena práctica que determina en gran medida la calidad de las aplicaciones de software. Es una solución documentada, que se ha aplicado con éxito en múltiples ambientes para erradicar problemas comunes de diseño de software, con una probada efectividad y con características de reutilización (34). En el sistema se evidencian los siguientes patrones:

**Instancia Única:** Instancia única o *singleton* por su nombre en inglés es una solución básica de arquitectura de software. Como patrón de creación a nivel de objeto proporciona una forma de agrupar el código de una unidad lógica, a la que se puede acceder haciendo uso de una única variable, garantizando la unicidad de esta instancia y ofreciendo un punto de acceso global a esta variable (17). En muchos proyectos web se utiliza espacios de nombre para reducir un número de variables globales, además de contribuir a la ramificación y acceso a funciones públicas.

En la capa controladora este patrón se implementa mediante la clase *Controller*, desde la cual heredan todos los controladores. La otra instancia única es el Enrutador, que contiene el método `getInstance ()` que ejecuta todas las peticiones que se realizan a la aplicación a través del controlador `sfWebFrontController`. Mientras que la variable global *sky*, se comporta como instancia única del sistema que se encarga de construir todas las clases, las funciones y acciones dentro de las vistas generadas por ExtJS.

**Comando:** Es un patrón de comportamiento enfocado en el manejo de menú desde las interfaces. Dado que un menú puede contener una serie de operaciones relacionadas con un tema concreto, la forma de invocar dichas funciones puede constituir un problema que afecte la reutilización y optimización del sistema. Este patrón se centra en la capacidad de configuración y llamadas a métodos, en dependencia de las veces que se necesite. Esto permite desacoplar los objetos de la acción o evento que las desencadenan, pudiendo separarse las interfaces de la lógica, elevando el grado de flexibilidad de las aplicaciones (34). Entre sus principales ventajas se pueden destacar que:

- Facilitar la parametrización de las acciones a realizar.
- Independizar el momento de petición del de ejecución.
- Implementar `Callbacks`, especificando que comandos queremos que se ejecuten en ciertas situaciones de otros comandos. Es decir, un parámetro de un comando puede ser otro comando a ejecutar.
- Desarrollar sistemas utilizando comandos de alto nivel que se construyen con operaciones más sencillas (primitivas).

Este patrón se manifiesta a través de la clase `sfWebController`, encargada de establecer el módulo y la acción que se usará por cada petición del usuario.

**Decorador:** Un decorador es un patrón estructural a nivel de objetos con el propósito de añadir responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades (34). Su estructura básica puede verse de la siguiente forma:

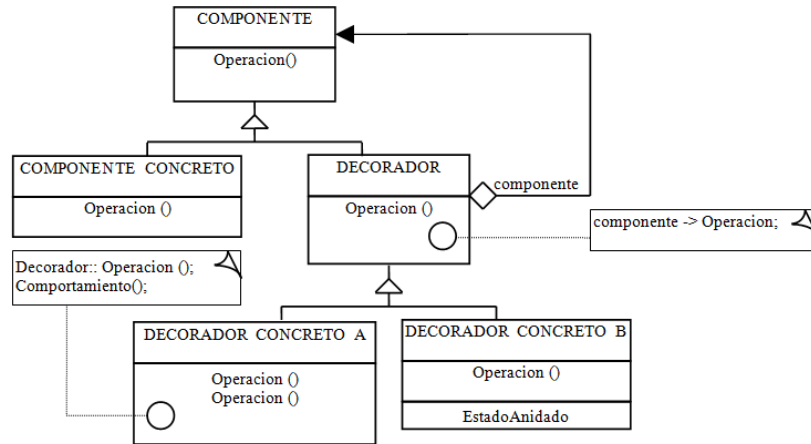


Figura 6. Vista del patrón Decorador

Es utilizado en la clase sfView, padre de todas las vistas y que permite agregar funcionalidades dinámicamente.

### 2.3.3 Diagrama de clases de diseño

El Diagrama de Clases es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones. Describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Además de los componentes que se encargarán del funcionamiento y la relación entre uno y otro (19).

A continuación, se muestran los diagramas de clase del análisis para los principales casos de uso del sistema.

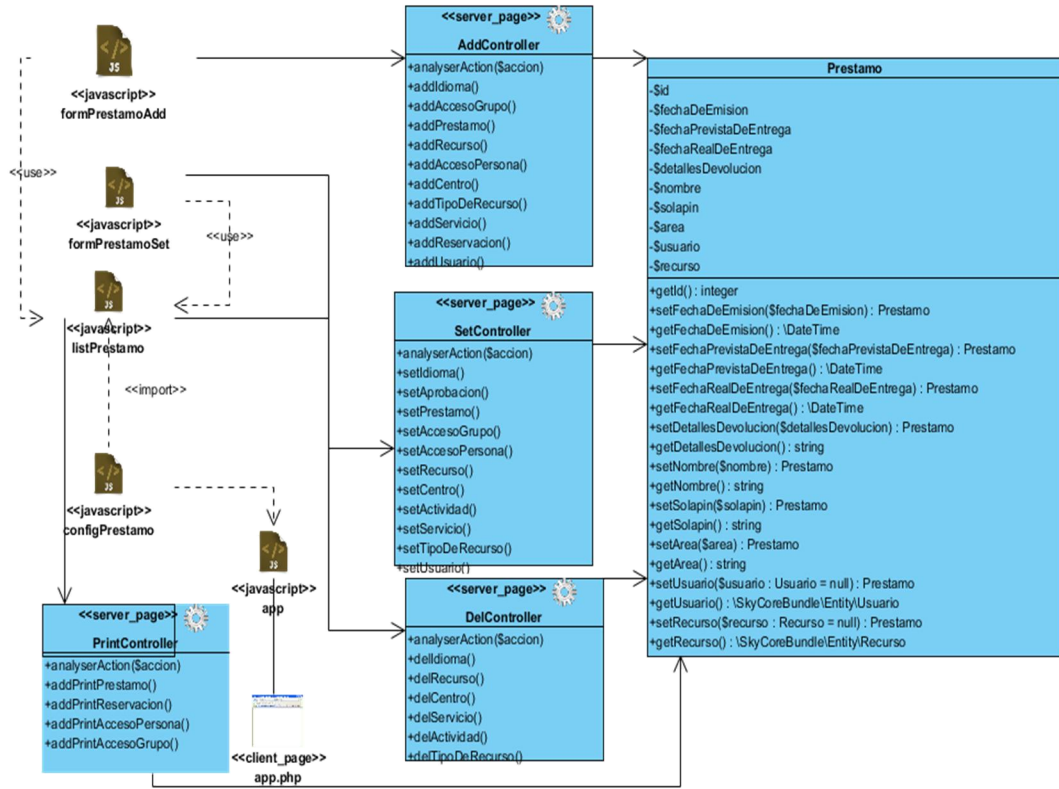


Figura 7. Diagrama de clases Caso de uso Gestionar préstamos.

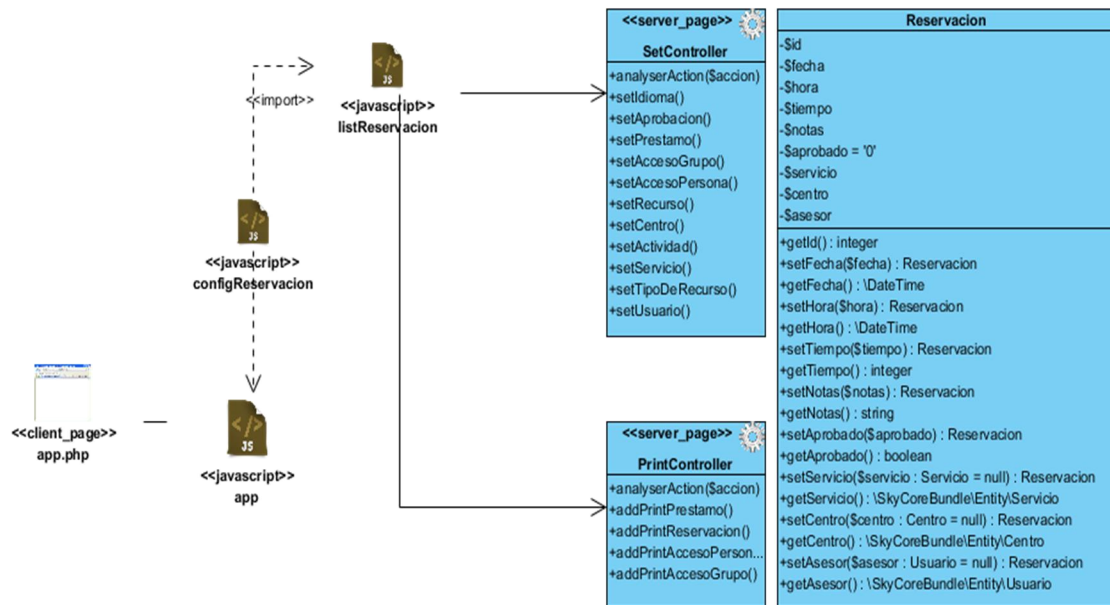


Figura 8. Diagrama de clases Caso de Uso Gestionar Reservación

### 2.3.4 Modelo de datos

El modelo de datos está descrito por el diagrama entidad-relación, que denota el modo físico en que está dispuesta la información dentro de la base de datos.

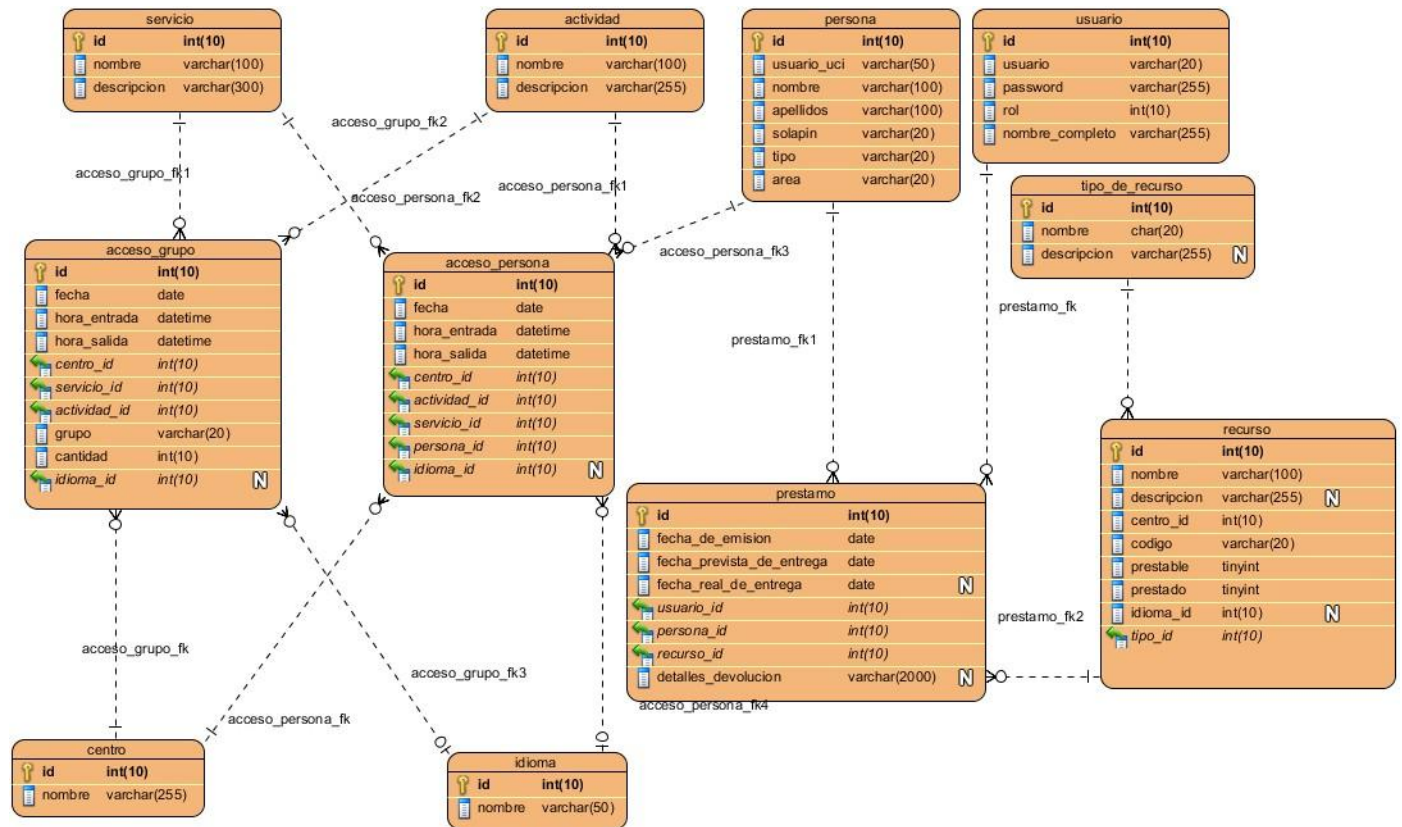


Figura 9. Diagrama Entidad-Relación

La descripción de las entidades es la siguiente:

Entidad	Descripción
Servicio	Registra los diferentes tipos de servicio que brinda el centro de idiomas.
Actividad	Registra las actividades que pueden desarrollarse en cada CASIE.
Centro	Registra los CASIE y otros centros destinados al usuario que tiene el centro de idiomas.
Idioma	Diferentes idiomas soportados para actividades, servicios y recursos.

Tipo de recurso	Registra las diferentes categorías de los recursos disponibles.
Recursos	Registra los recursos que tiene el centro de idiomas destinados a los usuarios.
Acceso de Persona	Para el registro de actividad de todas las personas que acceden a los CASIE
Acceso de Grupo	Para el registro de la actividad de los grupos docentes que acceden a los CASIE
Préstamo	Registra los préstamos que se efectúan sobre los recursos en cada CASIE
Usuario	Registra los diferentes usuarios del sistema.

*Tabla 7. Descripción de las entidades de la base de datos.*

### **2.3.5 Diagrama de despliegue**

El Diagrama de Despliegue muestra un conjunto de nodos y sus relaciones. Se describe la arquitectura física del sistema durante la ejecución, en términos de: procesadores, dispositivos y componentes de software y se detalla la topología del sistema: la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Es un objeto físico en tiempo de ejecución que representa un recurso computacional que generalmente tiene memoria y capacidad de procesamiento.

Para la comprensión del ambiente donde se desplegará la solución se propone el siguiente diagrama:

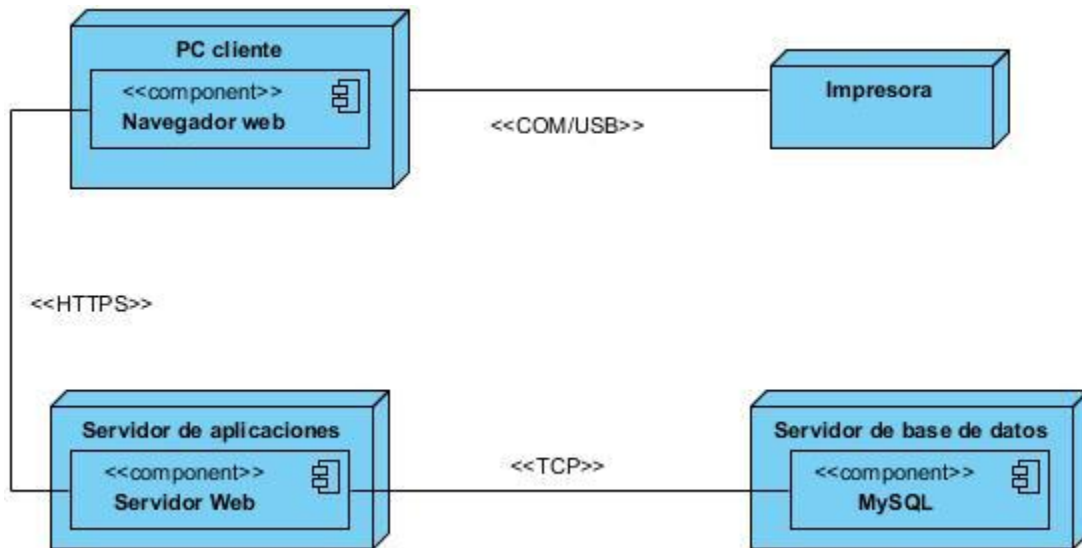


Figura 10. Diagrama de despliegue

## 2.4 Conclusiones parciales

Concluido el capítulo de Análisis y Diseño del sistema se obtienen las siguientes conclusiones: La identificación de las características y reglas del negocio permitió establecer el flujo de procesos y las condiciones a tener en cuenta durante la gestión de información en el sistema a desarrollar. Las definiciones de los 16 requisitos funcionales permitieron establecer las funcionalidades necesarias para cubrir las necesidades de automatización analizadas en la problemática. La elección de la arquitectura basada en el patrón Modelo-Vista-Controlador asegura una aplicación robusta, mejorable y adaptable a futuros requisitos.

## Capítulo 3 Implementación y pruebas

### Introducción

En el siguiente capítulo se abordará la etapa de implementación a partir de la definición del estándar utilizado y detallando el sistema mediante el modelo de implementación. Además, se realizará la validación del sistema a partir de las pruebas de software y se argumentarán los resultados del despliegue experimental de la solución propuesta.

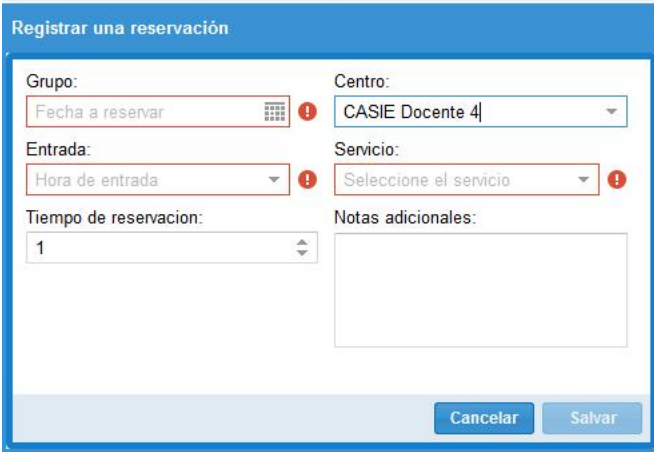
### 3.1 Modelo de implementación

Un modelo de implementación es un conjunto de artefactos que permiten documentar la construcción de un software en su etapa de codificación, a partir de la colección de componentes y subsistemas de implementación que lo conforman. Describe tanto los elementos del modelo de diseño, como las clases y otros artefactos que se comportan como componentes dentro del sistema, estos pueden incluir ejecutables, ficheros de código fuente, script y otros elementos necesarios para la implementación y el despliegue.

#### 3.1.1 Tratamiento de errores

EL proceso de tratamiento de errores tiene como objetivo asegurar la robustez y la corrección del sistema. Este objetivo se cumple mediante un sistema escalonado que funciona con la siguiente estructura.

Tratamiento de errores en las vistas: Durante la entrada de datos se validan mediante acciones JavaScript que no existan campos obligatorios vacíos. Además, se valida el formato para los datos de fecha, hora, listas de selección y números. En todos los casos se notifica al usuario mediante un mensaje de error en cada campo indicando las opciones de solución.



Registar una reservación

Grupo:	Centro:
Fecha a reservar	CASIE Docente 4
Entrada:	Servicio:
Hora de entrada	Seleccione el servicio
Tiempo de reservación:	Notas adicionales:
1	

Cancelar Salvar

Figura 11. Validación de datos de entrada en las vistas



Tratamiento de errores en el controlador: En el controlador se realiza una validación de formatos e integridad de datos durante la captura de información de desde la base de datos. En las acciones de registro y modificación se asegura que los datos tengan la estructura definida para cada campo de la base de datos. Se validan los formatos para los números, fechas, horas, y la composición de objetos. Se asegura también la presencia de campos no repetidos para asegurar la integridad de los datos.

### **3.1.2 Estándar de codificación**

Para asegurar un desarrollo bajo un estándar único y comprensible se establecieron las siguientes reglas:

1. La estructura de archivos de las vistas tiene la carpeta containers para los contenedores de cada acción, browsers para las listas y forms para los formularios.
2. Todas las listas de datos en las vistas tienen la estructura listObjeto. Ejemplo: listIdioma, listCentro.
3. Los formularios para el registro y modificación de datos tienen el formato formObjetoAdd y formObjetoSet respectivamente. Ejemplo: formCentroAdd y formCentroSet.
4. Todas las rutas generadas son procesadas por el método analyserAction.
5. Todas las acciones de registro son procesadas por el controlador addController.
6. Todas las acciones de modificación están procesadas por el controlador setController.
7. Todas las acciones de eliminación se procesan en el controlador delController.
8. Las acciones de impresión de datos se procesan en el controlador printController.

### **3.1.3 Diagrama de componentes**

El Diagrama de Componentes es la vista de implementación estática del diseño de un sistema .Se describe la estructura del software mostrando la organización y las dependencias entre un conjunto de componentes con sus interfaces y estructura interna.

A continuación se muestra el diagrama de componentes del sistema:

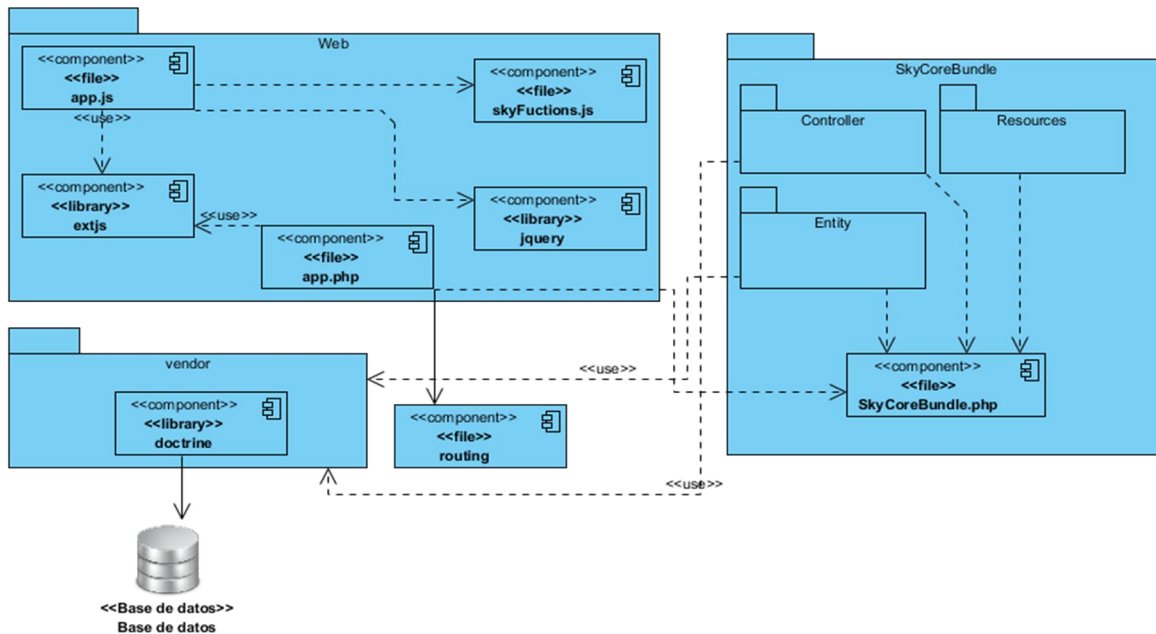


Figura 12. Diagrama de componentes

### 3.2 Pruebas de software

Las pruebas son actividades que se realizan para garantizar la calidad de las aplicaciones de software a partir del testeo de la misma o sus componentes. Las pruebas representan la revisión final de las especificaciones del diseño y la codificación. El resultado de las pruebas determina si el software cumplió con los requisitos establecidos durante su definición (18).

Para la aplicación se realizaron pruebas funcionales de caja negra, consistentes en la validación de entradas y salidas de cada una de las funcionalidades. Los niveles seleccionados para las pruebas son las pruebas de sistemas, consistente en la realización de pruebas al software como un todo para comprobar su comportamiento, lo que permitirá realizar la verificación de las funcionalidades de forma integrada y capturar los errores tal y como se presentarán luego del despliegue. Las pruebas fueron divididas en tres iteraciones.

En la primera fase de pruebas del sistema, a partir del análisis de cada funcionalidad se encontraron No Conformidades. Las mismas se distribuyeron según las categorías de Seguridad, Gestión de Acceso y Control de Servicios, y según el tipo de error que podía ser de validación, registro, obtención de información o informes. Los resultados fueron los siguientes:

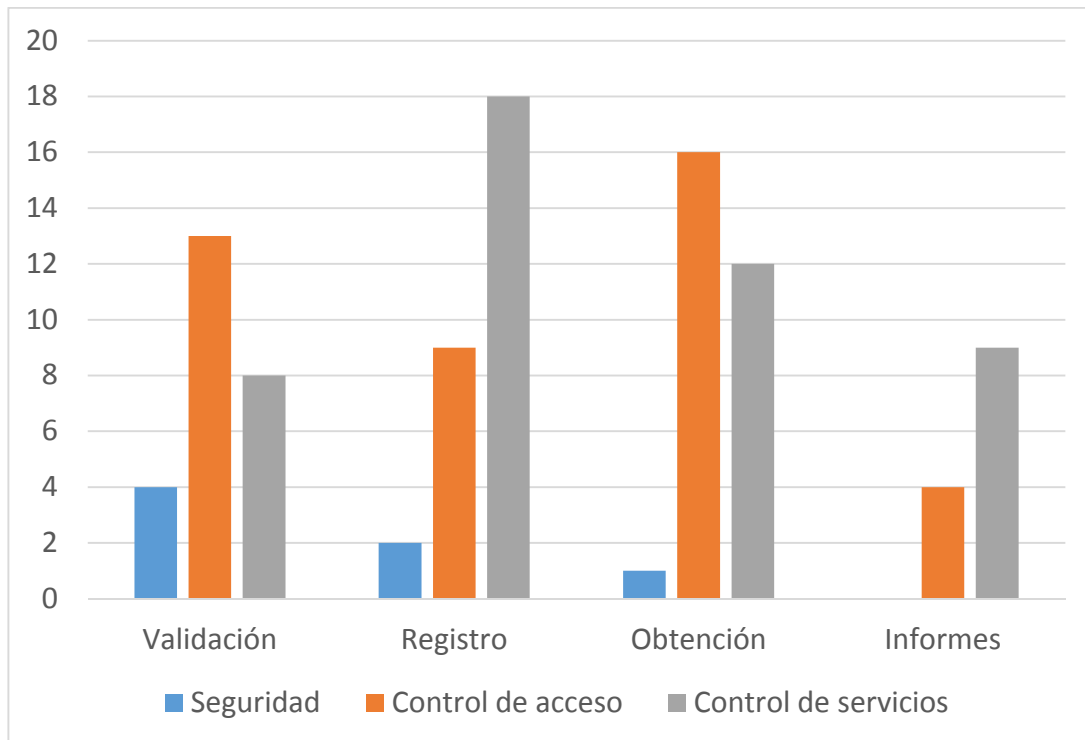


Figura 13. Registro de No Conformidades

Las No Conformidades fueron erradicadas y corregidas a partir de una estrategia de pruebas establecidas.

En la realización de las pruebas se utilizan los siguientes nomencladores en el juego de variables: V (válido), I (Inválido), N (Campo vacío), N/A (Sin utilización de variables de entrada)

A continuación, se evidencian los resultados de las pruebas utilizando la técnica de partición de equivalencia a los principales casos de uso del sistema Autenticar usuario, Gestionar recurso y Gestionar préstamo:

### 3.2.1 Pruebas para el caso de uso Autenticar Usuario

Sección de prueba	Escenarios a probar	Descripción
<b>Autenticar usuario</b>	Datos de usuarios correctos	Se introducen los datos de acceso correctos al sistema
	Usuario o contraseña incorrecto	Se introduce una combinación de usuario y contraseña inválidos.

Tabla 8. Secciones a probar Caso de uso Autenticar usuario

No	Nombre del campo	Clasificación	Nulo	Descripción
----	------------------	---------------	------	-------------

1	Usuario	Texto	No	Nombre del usuario UCI
2	Contraseña	Texto	No	Contraseña UCI del usuario

Figura 14. Variables del caso de Uso Autenticar Usuario

Escenario	Juego de variables	Respuesta del sistema	Resultado
Datos de usuario correctos	V,V	Registra al usuario conectado y accede al sistema	Satisfactorio
Usuario o contraseña incorrecto	V,I o I,V o I,I	Muestra un mensaje en una ventana flotante: "Combinación de usuarios y contraseña incorrecta."	Satisfactorio

Tabla 9. Matriz de pruebas de datos caso de uso autenticar usuario

### 3.2.2 Pruebas para el caso de uso gestionar Recursos

Sección de prueba	Escenarios a probar	Descripción
Adicionar recurso	Datos de recurso correctos	Se introducen correctamente todos los datos del recurso
	Campos vacíos	Se dejan sin llenar los campos de centro y tipo de recurso.
Modificar recurso	Recurso sin seleccionar	Se procese a Modificar sin seleccionar u recurso
Eliminar recurso	Eliminar recurso	Se procede a eliminar un recurso.

Tabla 10. Secciones a probar Caso de uso Gestionar recurso

No	Nombre del campo	Clasificación	Nulo	Descripción
1	Código	Texto	No	Código del recurso según el registro de activos fijos.
2	Nombre	Texto	No	Nombre con que se identifica al recurso.
3	Centro	Lista desplegable	No	Centro al que se encuentra asignado el recurso.

4	Idioma	Lista desplegable	No	Idioma relacionado con el recurso.
5	Tipo	Lista desplegable	No	Clasificación del recurso según su tipo
6	Prestable	Checkbox	Si	Se marca en caso de que el recurso pueda ser prestado.
7	Descripción	Texto	Si	Permite adicionar notas especiales sobre el recurso.

Tabla 11. Variables del caso de Uso Gestionar recurso

Escenario	Juego de variables	Respuesta del sistema	Resultado
<b>Datos de recurso correctos</b>	V,V,V,V,V,V,V	Registra un nuevo recurso en el sistema	Satisfactorio
<b>Campos vacíos</b>	V,V,N,V,N,V,V	Muestra un mensaje sobre los campos vacíos indicando que estos son obligatorios.	Satisfactorio
<b>Recurso sin seleccionar</b>	N/A	El sistema muestra un mensaje: "Debe seleccionar un recurso"	Satisfactorio
<b>Eliminar recurso</b>	N/A	El recurso queda eliminado del sistema.	Satisfactorio.

Tabla 12. Matriz de pruebas de datos

### 3.2.3 Pruebas para el caso de uso Gestionar préstamos

Sección de prueba	Escenarios a probar	Descripción
<b>Adicionar préstamo</b>	Datos del préstamo correctos	Se introducen correctamente todos los datos del préstamo
	Campos vacíos	Se dejan campos vacíos en el formulario de registro.
	Recurso prestado	Se selecciona un recurso que ya se encuentra prestado

	Recurso no prestable	Seleccionado un recurso que no puede ser prestado
<b>Entregar préstamo</b>	Entregar préstamo	Se procede a cerrar un préstamo

Tabla 13. Secciones a probar Caso de uso Gestionar préstamo

No	Nombre del campo	Clasificación	Nulo	Descripción
1	Nombre	Texto	No	Nombre y apellidos de la persona a la cual se le realiza el préstamo.
2	No. de solapín	Texto	No	Número de solapín de la persona.
3	Recurso	Lista desplegable	No	Recurso que se desea prestar
4	Área	Texto	No	Área a la que pertenece la persona que solicita el préstamo.
5	Fecha de entrega	Fecha seleccionable	No	Fecha en la que el usuario debe cerrar el préstamo.

Tabla 14. Variables del caso de Uso Gestionar préstamo

Escenario	Juego de variables	Respuesta del sistema	Resultado
<b>Datos del préstamo correctos</b>	V,V,V,V,V	Registra un nuevo préstamo en el sistema	Satisfactorio
<b>Campos vacíos</b>	V,N,N,N,V	Muestra un mensaje sobre los campos vacíos indicando que estos son obligatorios.	Satisfactorio
<b>Recurso prestado</b>	V,V,V,V,V	El sistema muestra un mensaje: "El recurso seleccionado no puede ser prestado"	Satisfactorio
<b>Recurso no prestable</b>	V,V,V,V,V	El recurso queda eliminado del sistema.	Satisfactorio.

<b>Entregar préstamo</b>	N/A	El sistema cierra el préstamo seleccionado	Satisfactorio
--------------------------	-----	--	---------------

Tabla 15. Matriz de pruebas de datos Gestionar préstamo

### 3.2.4 Pruebas para requisitos no funcionales

Las pruebas para los requisitos no funcionales permiten evaluar si los parámetros establecidos por cada una de las características deseables del sistema se cumplen con la terminación de la solución.

**Pruebas de usabilidad:** La usabilidad como atributo de calidad determina si el producto puede usarse por el conjunto de atributos para el cual fue concebido, de forma eficiente, efectiva y satisfactoria. Para ello se escogieron 4 técnicos de los CASIE y se establecieron las siguientes reglas:

1. Los usuarios deberían realizar las tareas comunes que generalmente realizan en la gestión de información de los CASIE.
2. Los usuarios deberán ser futuros usuarios del sistema.
3. Las pruebas se realizan en escenarios reales.

Se establecieron las escalas de: Usable, Medianamente usable y No Usable.

Los resultados arrojaron que 3 de los 4 usuarios determinaron el sistema Usable y 1 Medianamente usable, determinando como satisfactoria la prueba de usabilidad.

**Pruebas de Seguridad:** Para probar la seguridad del sistema se ejecutaron los siguientes casos de prueba:

1. Realización de ataques mediante Inyección de Código (sqli por las siglas del inglés *Structured Query Language Injection*). Se ejecutaron 20 ataques sqli siendo todos satisfactorios.
2. Fuerza bruta durante la autenticación de usuario. Se ejecutaron ataques secuenciales para forzar la autenticación de usuario. Los ataques se realizaron desconectados del servidor LDAP UCI y los resultados fueron satisfactorios.

**Pruebas de Interfaces externas:** Para las interfaces externas se validó la opinión de los 4 usuarios encuestados sobre las combinaciones de colores y distribución utilizadas. Los 4 usuarios aprobaron el diseño utilizado. Para comprobar el ajuste de las interfaces a distintas resoluciones se accedió a la aplicación desde un dispositivo móvil de 5 pulgadas y desde una tableta de 10 pulgadas, además de su prueba en monitores de 15, 18 y 21 pulgadas. Para todos los casos el resultado fue Medianamente Ajustable.

**Pruebas de Rendimiento en Hardware:** Las pruebas se realizaron en un ordenador servidor y cliente tal y como se establece en los requisitos no funcionales. Para la carga y el estrés se utilizó la herramienta JMeter, corriendo una prueba para 300 conexiones simultáneas los resultados fueron:

	Muestra	Media	Mediana	Línea 90%	Min	Max	Error	Rendimiento	Kb/seg
Total	5335	7710	110	18.985	0	415 K	0.009	4.090 /seg	80022.7

Tabla 16. Resultados de la prueba con el JMeter

Esto significa que la aplicación puede soportar hasta 5335 conexiones simultáneas, por lo que puede operar con un máximo de 55 usuarios concurrentes y con más de 70 no responde adecuadamente a las peticiones. El tiempo de recuperación es de 60 segundos. El por ciento de errores está representado por la carga de la imagen de la interfaz principal. Los datos obtenidos son adecuados y permiten que se cumplan los requisitos de rendimiento.



Figura 15. Resultados de prueba con JMeter

### 3.2.5 Validación de resultados obtenidos

Para validar los resultados de la investigación se efectúa los criterios establecidos en la introducción de: Un sistema de gestión para los CASIE que permita la automatización del control de acceso y los servicios brindados por el Centro de Idiomas de la Universidad de las Ciencias Informáticas.



Las opiniones de los 4 usuarios encuestados fueron:

Criterio	Opinión favorable	Resultado
Control de acceso	4 de 4	Satisfactorio
Gestión de préstamos	3 de 4	Satisfactorio
Gestión de reservaciones	4 de 4	Satisfactorio
Seguridad	4 de 4	Satisfactorio
Informes gráficos	2 de 4	Poco satisfactorio
Informes PDF	3 de 4	Satisfactorio

Tabla 17. Criterios para la validación de la solución por el usuario

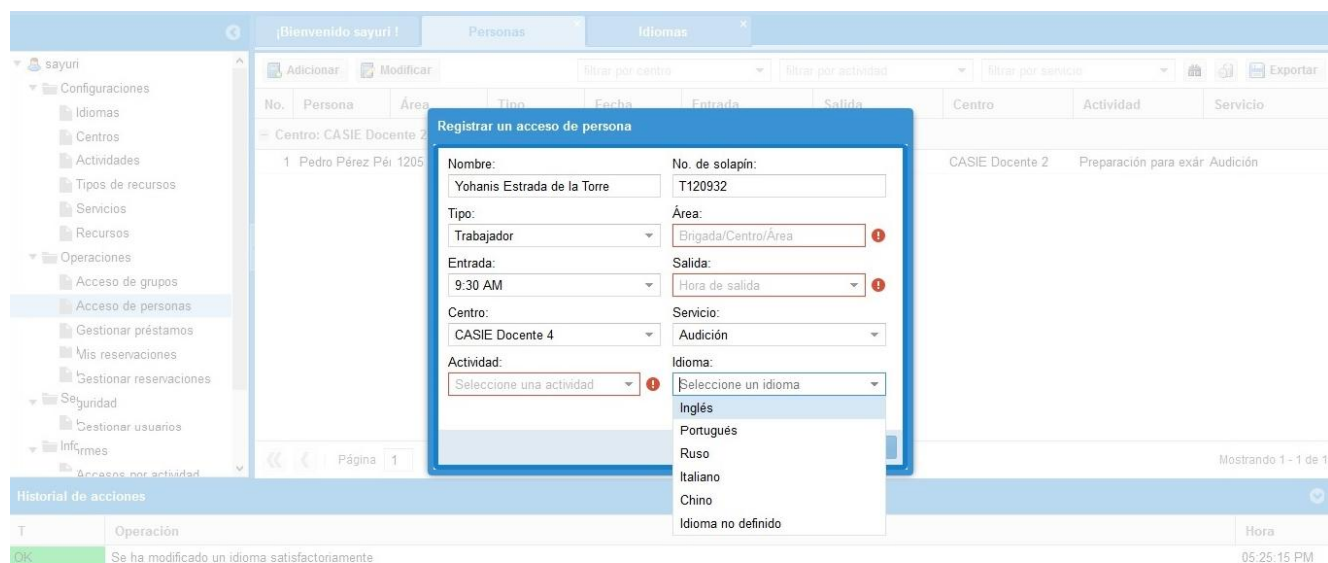


Figura 16. Registro de acceso de personas

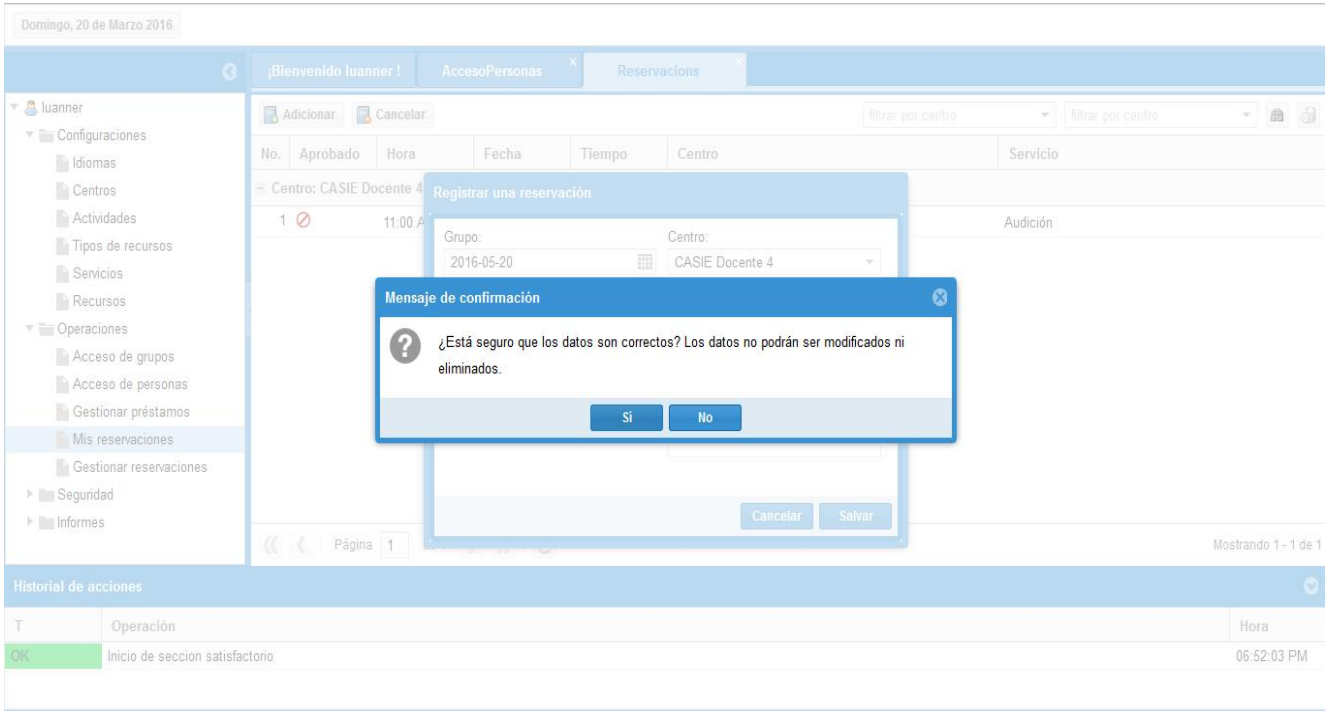


Figura 17. Registro de reservaciones

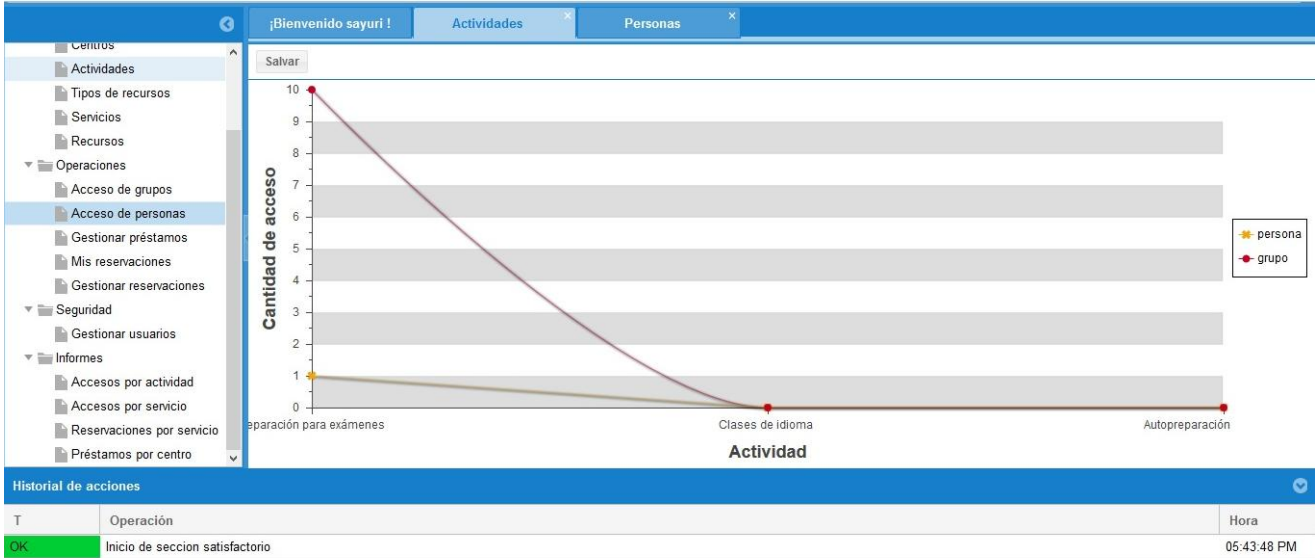


Figura 18. Informe de acceso por actividad

Para consultar otras Interfaces del Sistema ver el [Anexo no. 2](#) y [Anexo no. 3](#).

### **3.3 Conclusiones parciales**

A partir de la implementación y las pruebas se obtienen las siguientes conclusiones: Los procedimientos de validación de errores definidos permitieron asegurar la corrección y robustez de la aplicación, lo que mejorará la experiencia de usuario y la fiabilidad del sistema. El estándar de codificación empleado permite que la aplicación sea fácil de mantener, documentar y dar soporte. Las pruebas de funcionalidad mediante método de caja negra a partir de la partición equivalente corroboraron que las funcionalidades del sistema responden satisfactoriamente, al igual que los requisitos no funcionales tenidos en cuenta. Las validaciones del sistema a partir por parte de los usuarios determinaron el cumplimiento de los objetivos planteados para la investigación.

## Conclusiones

Se establecen las siguientes conclusiones generales:

1. La realización del sistema de gestión de información de acceso y uso de servicios para los CASIE del Centro de Idiomas de Universidad de las Ciencias Informáticas permitirá mejorar el proceso de planificación y análisis, lo que contribuirá a la toma de decisiones.
2. El sistema desarrollado optimizará las condiciones de trabajo de los técnicos generales de los CASIE, reducirá los gastos de tiempo, materiales empleados en la gestión de información y admitirá a la dirección hacer un seguimiento de los procesos de control de acceso y servicios de una forma rápida y precisa.
3. Los reportes realizados por el sistema, así como los datos almacenados, brindan un registro histórico del funcionamiento del Centro de Idiomas, asegurando una base de datos fiable para el análisis de información e influir positivamente en la adaptabilidad de los CASIE a los nuevos métodos de aprendizaje basados en sistemas de información y las tecnologías.

## Recomendaciones

Se recomienda lo siguiente:

1. Extender el sistema con funcionalidades dirigidas a los usuarios que consumen los servicios del Centro de Idiomas de la Universidad de las Ciencias Informáticas, como la gestión y publicación de certificaciones de idiomas y verificación de materiales de estudio.
2. Integrar la herramienta con el Sistema de Gestión Universitaria para asegurar funciones de información académica que mejorará la capacidad del sistema para el análisis estadístico.

## Citas

1. **Universidad Autónoma de México.** UNAM. [En línea] UNAM. [Citado el: 18 de Noviembre de 2015.] <http://dgb.unam.mx/index.php/servicios-bibliotecarios-y-de-informacion/servicios-bibliotecarios>.
2. **León, Universidad de.** Centro de Idiomas de la Universidad de León. [En línea] ULE. [Citado el: 20 de 11 de 2015.] <http://fgulem.unileon.es/ci/>.
3. **SEA.** Asociación de Centros de Idiomas. [En línea] SEA. [Citado el: 18 de 11 de 2015.] <http://www.idiomas.org.ar/index.php>.
4. **UNIR.** Idiomas de la Universidad de la Rioja. [En línea] Universidad de la Rioja. [Citado el: 24 de 11 de 2015.] <http://www.unir.net/idiomas/>.
5. *Gestión del conocimiento versus gestión de la información.* **Fernández Marcial, Viviana.** 41, Madrid : Scielo, 2006, Vol. 20. ISSN: 0187-358X.
6. **Gonzáles del Monte, María Isabel.** *Módulo de recuperación de información para el Sistema de Gestión de Documentos Históricos Dexscriba.* La Habana : Universidad de las Ciencias Informáticas, 2013. Tesis de grado.
7. **Fernando, Ale.** Monografías. [En línea] Monografías, 2010. [Citado el: 28 de Julio de 2014.] <http://www.monografias.com/trabajos89/introduccion-al-html-5/introduccion-al-html-5.shtml..>
8. **Blanco Espinoza, Lázaro.** *La Informática en la dirección de empresas.* La Habana : Félix Varela, 2011.
9. **Universidad de Entre Ríos.** Portal de proyecto Open Marcopolo. [En línea] UNER, 2003. [Citado el: 21 de Noviembre de 2015.] <http://marcopolo.uner.edu.ar/index.htm>.
10. **Proeycto Koha.** Proyecto Koha. [En línea] Universidad de La Plata. [Citado el: 20 de Noviembre de 2015.] <http://koha.unlp.edu.ar/>.
11. **Fonbellina, Javier.** *Sistema de gestión y control de préstamos de libros de biblioteca para teléfonos móviles andorid.* Madrid : Universidad Carlos III de Madrid, 2012.
12. **Ferreiro, Dariela y Elejalde, Elayne Yanet.** *Langlish: Sistema de apoyo a la autogestión del conocimiento en los CASIE.* La Habana : Universidad de las Ciencias Informáticas, 2012. Trabajo de Diploma.

13. **Griñán Furones, Iván.** *Sistema para la gestión de recursos y servicios de los centros de autoaprendizaje de idiomas extranjeros. Versión 2.* La Habana : Universidad de las Ciencias Informáticas, 2012.
14. **Duggan, Evan y Reichgelt, Han.** *Measuring Information Systems Delivery Quality.* Hershey. s.l. : Idea Group Inc., 2012. ISBN 1-59140-859-8.
15. **Krcuten, P.** *The Rational Unified Process, An Intruduction.* s.l. : Addison Wesley, 2004. ISBN: 978-032-1197-70-2.
16. **Pressman, Roger.** *Ingeniería de Software: un enfoque práctico.* s.l. : Mc Graw Hill, 2008. ISBN: 970-10-5473-3.
17. **Brooks, Frederick Phillips.** *Prácticas de Software.* [En línea] 2011. [Citado el: 11 de Agosto de 2014.] <http://www.practicadesoftware.com.ar>.
18. **Addison Wesley object technology series.** *Softwre development for small team: A RUP - centric approach.* s.l. : Addison Wesley, 2004. ISBN: 978-0321-199-50-8.
19. **Fowler.** *UML distilled, A brief guide to standard object modeling lenguaje.* Tercera. s.l. : Pearson Education, 2007. ISBN: 978-8131-715-65-9.
20. **McConell, S.** *Professional Software Development.* s.l. : Addison Wesley, 2003. ISBN : 0-321-19367-9.
21. **Meyer, B.** *Construcción de Software Orientado a Objetos.* La Habana : Félix Varela, 2009.
22. W3C. [En línea] [Citado el: 15 de 20 de 2015.] <http://www.w3.org/TR/websockets/>.
23. **Merley, Jimi.** *¿Por qué elegir PHP?* 2012.
24. **Pérez, Javier.** *¿Por qué JavaScript?* s.l. : La web del programador, 2014. Blog.
25. **De los Ángeles, O.** *Marcos de Trabajo para el desarrollo de Aplicaciones Web.* 2012.
26. **WebDesignish.** WebDesignish. [En línea] WebDesignish. [Citado el: 21 de 11 de 2015.] <http://www.webdesignish.com/>.
27. **Eguiluz, Javier.** *Desarrollo Ágil con Symfony 2.* 2013.
28. **Ext JS Inc.** Ext JS. [En línea] [Citado el: 11 de noviembre de 2015.] [www.extjs.com](http://www.extjs.com).
29. **Arregoces, Mauricio y Portolani, Maurizio.** *Data Center Fundamentals.* Indiana : Cisco Press, 2011. ISBN: 1-58705-023-4.
30. **Domínguez , Dorado.** *NetBeans IDE 4.1, la alternativa a Eclipse.* Madrid : Iberprensa, 2005.

31. **Visual Parading Group.** *Reasons to Choose Visual Paradigm.* s.l. : Visual Parading Group, 2011.
32. **ASF.** Apache Software Foundation. [En línea] Apache Software Foundations. [Citado el: 22 de Noviembre de 2015.] <http://www.apache.org/>.
33. **Potencier, Fabien.** *Symfony 2.8.3 Reference Guide.* EE.UU : libroweb, 2015.
34. **Fowler, Martin, Rice, David y Foemmel, Matthew.** *Patterns of Enterprise Application Architecture.* s.l. : Addison Wesley, 2002. ISBN : 0-321-12742-0 .
35. **Jacobson, Ivar, Booch, G. y Rumbaugh, J.** *El proceso unificado de desarrollo de software.* Madrid : Pearson Educación, 2000. ISBN: 84-7829-036-2.
36. **CEEC.** *Informe Central.* La Habana : CEEC, 2012.
37. **Espinosa Fuentes, Fernando.** *Sistema de informacion para la gestion de empresa.* s.l. : Universidad de Talca, 2013.
38. **Gerner, Jason.** *Professional LAMP: Linux, Apache, MySQL and PHP 5 Web Development.* Indianapolis : Wiley Publishing, 2011. SBN: 978-0-7645-9723-7.
39. **Laudon, Lane y Laudon, Kenneth.** *Sistemas de información gerencial- Administración de la empresa digital.* s.l. : Prentice Hall : Pearson Education, 2006.
40. **Strouptus, VB.** *The C++ Programming Language (6th Edition ed.).* New Jersey : Addison Wesley, 2011.
41. **van der Graaf, Shenja.** *Information Communication Technologies and Emergin Business Strategies.* London UK : Idea Group Inc, 2007. ISBN 1-59904-236-3.
42. **Wallace, Doug y Ragget, Isobel.** *Extreme Programming for Web Projects.* s.l. : Addison Wesley, 2002. ISBN : 0-201-79427-6.
43. **PostgreSQL Development Group.** *PostgreSQL 9.3.0 Documentation.* s.l. : PostgreSQL Global Development Group, 2013. Manual.
44. **Bohem, Barry y Turner, Richard.** *Balancing Agility and Discipline: A Guide for the Perplexed.* Boston : Pearson Educations Inc., 2004. ISBN: 0-321-18612-5.
45. **OMG.** *Unified Modeling Language Especification.* Tercera. s.l. : Pearson Education,, 2011.
46. **Quesada, Antonio.** *Pruebas de Software.* 2011.
47. **Larman, Craig.** *UML y Patrones.* s.l. : Prentice Hall, 2003.



48. **ITSON, Biblioteca.** [En línea] 2014. [Citado el: 3 de diciembre de 2014.] [http://biblioteca.itson.mx/oa/dip\\_ago/introduccion\\_sistemas/p3.htm](http://biblioteca.itson.mx/oa/dip_ago/introduccion_sistemas/p3.htm).
49. **Gómez, A y Wilson, J.** Metodología de desarrollo de software un enfoque práctico y global versión 1.0.11. [En línea] 23 de marzo de 2011. <http://www.otcolombia.com/documentos/mds360-1.0.11-beta.pdf...>
50. **Bent, Kent y Fowler, Martin.** *Planning Extreme Programming*. s.l. s.l. : Addison Wesley, 2001.
51. **Frias, C.** *Desarrollo de una librería de componentes de interfaz gráfica de usuario basada en Swing Application Framework y AWT Framework para el desarrollo de la capa de presentación en aplicaciones de software de escritorio*. La Habana, UCI : s.n., 2012.
52. **Potencier, F.** *Symfony 1.2, la guía definitiva*. 2010. ISBN-13: 978-1590597866..
53. **Dans, Enrique.** Information Management. Reflexiones sobre las tecnologías de la información. [En línea] 28 de noviembre de 2006. [Citado el: 7 de septiembre de 2014.] <http://informationmanagement.wordpress.com/category/gestion/gestion-de-la-informacion/...>
54. **Institute, The British Standards.** The British Standards Institute. [En línea] , 2013. [Citado el: 10 de septiembre de 2014.] <http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion>.
55. **Estrada, Salvador y Sabando, David.** *Gestión de recursos tecnológicos. Doctorado Interuniversitario en Economía y Gestión de la Innovación y Política Tecnológica*. Madrid, España : s.n., 2001.
56. **Sánchez Muguercia, Nilmar.** *Aplicación de escritorio del Sistema de Entrega Digital*. . La Habana : Universidad de las Ciencias Informáticas, 2013.
57. **Baisley, B.** *Unified Modeling Language Infrastructure*. s.l. : Pearson Editorial, 2006.
58. **Poyo Garrido, Alejandro.** Comparativa Framework. [En línea] 2012. [Citado el: 26 de octubre de 2014.] <http://es.scribd.com/doc/189203494/Comparativa-Framework..>
59. **Mena, D.** Los IDE de desarrollo. [En línea] 2011. <http://www.cursosporinternet.info/index.php/the-news/43-programacion/167-los-ides-de-programacion.html> ..
60. **Álvarez, M.** Un IDE para el desarrollo de aplicaciones web, enfocado en Ajax y la Web 2.0. [En línea] 2013. <http://www.desarrolloweb.com/articulos/aptana-studio.html..>
61. **Crusat y Vaquer, Marc.** *Zend Studio: el entorno de desarrollo para PHP*. 2004.

62. **López Michelone, Manuel.** PhpStorm. [En línea] 2013. <http://www.unocero.com/2013/10/25/se-libera-phpstorm-8/>.
63. **Louden, Kenneth.** *Lenguajes de programación: Principios y práctica.* s.l. : Cengage Learning Editores, 2004.
64. **Ajona Prado, Guillermo.** Página Web sobre HTML y CSS. . [En línea] 2012. [Citado el: 26 de octubre de 2014.] <https://belenus.unirioja.es/~guprado/pagweb/caraccss.html>..
65. **Eguiluz Pérez , Javier.** *Introducción a javaScript.* 2009.
66. **Sæther Bakken, Stig.** *Manual de PHP.* s.l. : Free Software Foundation, 2001.
67. **Letelier.** *Metodologías ágiles en el desarrollo de software.* . Valencia : s.n., 2007.
68. **Rizo Constanten, Rosa María.** *Sistema de Comunicación interna. Tesis de grado para optar por el título de Ingeniera en Ciencias Informáticas.* La Habana : Universidad de las Ciencias Informáticas, 2013.
69. **Sánchez González, Carlos.** *ONess: un proyecto open source para el negocio textil mayorista desarrollado con tecnologías open source innovadoras.* Caruña : s.n., 2004.
70. **Winesett, Jeffery.** *Agile Web Application Development with Yii 1.1 and PHP 5.* . s.l. : Birmingham: Packt Publishing Ltd, 2010. ISBN 978-1-847199-58-4..
71. **Wallace.** *Extreme Programming for Web Projects.* s.l. : Addison Wesley, 2002. ISBN: 0-201-79427-6.
72. **Casas, Sandra y Reinaga, Héctor.** *Identificación y Modelado de Aspectos Tempranos dirigido por Tarjetas de Responsabilidades y Colaboraciones.* 2008. <http://www.oocities.org/espanol/profeprog2/INVPAPER25.pdf>..
73. **Real Academia de la Lengua Española.** Real Academia Española. [En línea] 2013. <http://lema.rae.es/drae/?val=informaci%C3%B3n>.
74. **Koskela, Lasse.** *TEST DRIVEN, Practical TDD and Acceptance TDD for Java Developers.* s.l. : Manning Publication Co, 2008. ISBN 1-932394-85-0.
75. **Fowler, Martín.** *Pattern Enterprise Application Architecture.* s.l. : Addison Wesley, 2002. ISBN: 0-321-12742-0.
76. **Pressman, Roger.** *Ingeniería de Software. Un enfoque práctico.* . 2008. ISBN 978-0-07-337597-7.
77. **Suehring, S.** *MySQL Bible.* New York : Wiley Publishing Inc, 2001. ISBN: 0-7645-4932-4.

78. **XAMPP**. [En línea] 2015. [Citado el: 16 de enero de 2015.] <https://www.apachefriends.org/es/index.html>.
79. **Davis, G. y Olsón**. *Management Information Systems: Conceptual foundations, Structure and Development*. Medellín : Universidad de Antioquia, 1985.
80. **Bent, Kent y Fowler, Martin**. *Planning Extreme Programming*. s.l. : Addison Wesley, 2001. ISBN: 0-201-71091-9..
81. **Gamma, Erich y otros**. *Design Patterns: Elements of Reusable Object-Oriented Software*. s.l. : KevinZhang, 1997.
82. **Guerrero, Z y otros**. *Sistema de Captura y Transcripción de Audio*. La Habana : Universidad de las Ciencias Informáticas, 2009.
83. **Jacobson y otros**. *El proceso unificado de desarrollo de software*. Madrid : Pearson Education, 2000. ISBN:84-7829-036-2.
84. **Jeffries, Ron y otros**. *Extreme Programming Installed*. s.l. : Addison Wesley, 2000. ISBN: 0-201-70842-6.
85. **Letelier, P. y otros**. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Universidad de Valencia. Valencia : Valencia : Universidad de Valencia : s.n., 2008. Vol. 05, 26, 2008. Vols. 05,26. ISSN 1666-1680.
86. **Pérez, Mauri y otros**. *Desarrollo del sistema de gestión de contenidos de la colección El Navegante en su versión multiplataforma*. La Habana : s.n., 2012.
87. **Rodríguez Cabrera, Maite y otros**. La Metodología XP Aplicable al Desarrollo del Software educativo en Cuba. [En línea] 2007. [http://repositorio\\_institucional.uci.cu//jspui/handle/ident/.TD\\_0837\\_07](http://repositorio_institucional.uci.cu//jspui/handle/ident/.TD_0837_07).
88. **Serafín Linares, Ana Mary y otros**. *SISTEMA AUTOMATIZADO PARA EL CONTROL Y GESTIÓN DEL TRANSPORTE EN EL GRUPO DE LA ELECTRÓNICA*. La Habana : Universidad de las Ciencias Informáticas, 2007. 2007.0841\_07.
89. **Tellez, I y otros**. *Sistema para la gestión de la información de los medicamentos, turnos médicos y vacunas en el hospital Ernesto Ché Guevara*. La Habana : s.n., 2012.
90. **Díaz Pérez, Maydelín y Contreras Rivero, Yimian**. Características de los sistemas de información que permiten la gestión oportuna de la información y el conocimiento institucional. [En línea] noviembre de 2009. [Citado el: 10 de septiembre de 2014.] <http://scielo.sld.cu/scielo.php?script=sc>. ISSN 1561-2880.

## Anexos

### Anexo no. 1: Encuesta

Saludos:

Como parte del desarrollo de un sistema de gestión para el Centro de Idiomas de la Universidad de las Ciencias Informáticas, estamos realizando esta sencilla encuesta. Sería de mucha utilidad que usted valorara los aspectos que mencionamos a continuación en las escalas Bien (B), Regular (R) y Mal (M), además de responder Si o No a ciertas preguntas. No tomaremos más de 2 minutos de su tiempo y los resultados son anónimos.

1. ¿Cómo considera el proceso de planificación de actividades en los CASIE?

Bien: \_\_\_\_\_ Regular: \_\_\_\_\_ Mal: \_\_\_\_\_

2. ¿Cómo evalúa el actual método de préstamos de recursos del Centro de Idiomas?

Bien: \_\_\_\_\_ Regular: \_\_\_\_\_ Mal: \_\_\_\_\_

3. ¿Cree importante que se puedan reservar los servicios a través de una aplicación?

No: \_\_\_\_\_ Si: \_\_\_\_\_

4. ¿Cree que es necesario un sistema de control de acceso y uso de servicios?

No: \_\_\_\_\_ Si: \_\_\_\_\_

**Muchas gracias, sus respuestas serán de gran utilidad para nuestra investigación.**

### Anexo no. 2: Interfaz Autenticar usuario



Usuario:

Contraseña:

### Anexo no. 3: Interfaz Registrar un Préstamo

The image shows a software interface titled "Registrar un Préstamo" (Register a Loan). It contains several input fields and two buttons. The fields are arranged in a grid-like structure:

- Nombre:** A text input field with the placeholder "Nombre y apellidos".
- No. de solapín:** A text input field with the placeholder "Solapín T#####".
- Recurso:** A dropdown menu with the placeholder "Seleccione un recurso".
- Área:** A text input field with the placeholder "Brigada/Centro/Área".
- Fecha de entrega:** A date picker input field with the placeholder "Fecha de entrega".

At the bottom right of the form, there are two buttons: "Cancelar" (Cancel) and "Salvar" (Save).