

# **Universidad de las Ciencias Informáticas. Facultad 2.**



**Título: Desarrollo de un Software Educativo para la Enseñanza y Aprendizaje de la Pronunciación del Inglés.**

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

**Autores:**

Jorge Javier Garcia Ruíz.  
Adrian Roberto Correa Borrego.

**Tutor:**

Ing. Maidelis Milanés Luque.

**Co\_Tutor:**

Lic. Abdel Eduardo Martínez Valverde.

Cuidad de la Habana, Cuba. Junio, 2009.

*No basta saber, se debe también aplicar. No es suficiente querer, se debe también hacer.*

*Johann Wolfgang Goethe*

## **DECLARACIÓN DE AUTORÍA.**

Declaramos que somos los únicos autores, y autorizamos a la facultad 2 de la Universidad de las Ciencias Informáticas; así como dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de junio del 2009.

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

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

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

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

## AGRADECIMIENTOS

*No resulta fácil ser breve a la hora de agradecerles a todas las personas que han estado a mi lado brindándome su ayuda y apoyo en todo momento durante estos 5 años de estudio. Agradezco a mis padres a los que admiro y quiero mucho. Mimi, papi, gracias por su apoyo y sabios consejos; un millón de gracias por darme la oportunidad de estar aquí en estos momentos, por hacerme ver el lado positivo de las cosas por muy difícil que fuese el camino, gracias por animarme siempre, por hacerme ver que con esfuerzo, mucha fe y dedicación todos salimos adelante, por estar siempre a mi lado brindándome su cariño en los momentos de felicidad y tristeza, gracias por saberme mostrar el camino correcto, por su educación y por ser ejemplos de padres. Gracias por su amor y afecto. Ustedes son los mejores padres del mundo... los quiero con la vida. A mi hermana querida quiero agradecerle por su apoyo brindado de una manera u otra, por estar siempre pendiente de mí y de mis estudios. Para ti mi hermana un beso y todo mi corazón. A mis tíos Juan Miguel Díaz y Mirta Borrego gracias por su ayuda valiosa y sin límites, y gracias también por nunca decir un "no" cuando los necesité, los quiero. A mi novia por su apoyo incondicional y paciencia, por ser un motivo más por el cual seguir adelante. A mis abuelos y tíos de La Habana, que aunque no están presentes se que están conmigo. Quiero agradecerle a toda mi familia por siempre estar pendiente de mí y de mis estudios, gracias por su apoyo y ayuda incondicional y por la confianza depositada. A mis amigos, a los que han estado a mi lado compartiendo días de estrés, alegrías y tristezas durante estos 5 años, a todos aquellos que han demostrado ser verdaderos amigos. A los profesores que me han brindado siempre su ayuda. A nuestro eterno Comandante en Jefe y a la Revolución, que me han dado la posibilidad de superarme profesionalmente. Quiero agradecer a nuestros tutores, y demás personas que de una forma u otra, por su colaboración y paciencia contribuyeron a la materialización de este sueño.*

*Gracias a todos.*

*Adrian R. Correa Borrego*

*A lo largo de todo este tiempo en la Universidad recibí ayuda de muchas personas, algunas tal vez no estén presentes hoy, pero aún así quiero agradecerles por este gran logro. Tengo que comenzar con mis padres que fueron insoportables hasta el último momento para que estudiara y me graduara. Mi mamá que no descansó ni un momento de interesarse por mis estudios y mi salud, a mi papá que siempre ha intentado hacer de mí un hombre honesto y trabajador y de quién he aprendido infinitos conocimientos. A mi hermanito que aún siendo menor que yo es mi ejemplo de superación profesional en la informática, sé que no tardarás en igualarme así que pronto será tú graduación la que estaremos presenciando... A mis abuelos que deben estar presentes en este día, porque apuesto que no se lo perderían por nada del mundo. A la señora que desde la distancia siempre apostó por mí y siguió mis progresos, a los Tomás que han estado a nuestro lado en momentos buenos y malos incondicionalmente. Al teacher que me dió los conocimientos del inglés para ganar tantos premios y graduarme hoy con una tesis que ayudará a otros a aprender, como lo hizo él una vez conmigo. A los profesores que me enseñaron que aún más valioso que títulos y calificaciones es el conocimiento que hoy me llevo conmigo. A Maidelís, profesora, tutora y amiga a quién tuve la suerte de conocer, gracias por empujarnos hasta la meta!!! A mis amigos de la UCI con quienes tengo una enorme cantidad de historias que nunca voy a olvidar, aún cuando la distancia de nuestro deber a cumplir nos fuerce a dejar de vernos. Ustedes estarán presentes en mi vida diaria. A mis amigos de la Habana cuya genialidad fue una continua fuente de inspiración y quienes celebran mis logros y no dejan de ayudarme en mis errores. Los que un día junto a mí en un lugar lejano en el pasado imaginaron conmigo este momento. A esas personas que no están en esta sala hoy, pero sin quienes este sueño hubiese sido imposible, a ustedes espero estén observando este momento y sonrían como yo, por otra meta lograda.*

*Gracias a todos.*

*Jorge Javier Garcia Ruíz*

## **DEDICATORIA**

*Dedicado a mis padres Mary y Roberto, quienes siempre han soñado con ver a su hijo convertirse en un profesional comprometido con la Revolución y útil a la sociedad, con este trabajo hago realidad su sueño porque se lo merecen y porque para eso se han sacrificado a lo largo de todos estos años ofreciéndome su apoyo y siendo mi retaguardia en el camino hacia el futuro. Mimi y papi los quiero infinitamente y de ustedes es este triunfo personal. Muchas Gracias.*

*Adrian R. Correa Borrego*

*Dedicado a mis padres Jorge y Caridad y a mi hermano Gabriel, este sueño es más de ustedes que mío. En agradecimiento por todo lo que han dado durante este tiempo suya es esta tesis. Que es sólo una pequeña parte de lo mucho que les debo.*

*Jorge Javier Garcia Ruíz*

## RESUMEN

En este trabajo se presenta el desarrollo de un sistema de apoyo para la enseñanza aprendizaje de la pronunciación inglesa en nuestra universidad. Atendiendo a las peticiones del cliente se decide implementar un juego en plataforma Web como propuesta didáctica para abordar esta problemática.

Para la elaboración de la aplicación se utilizó como gestor de base de datos PostgreSQL, lenguaje de programación PHP y Apache como servidor Web. Se decidió trabajar con la arquitectura cliente/servidor y el patrón Modelo Vista Controlador. También se hizo uso de la metodología de desarrollo RUP, del lenguaje de modelado UML y de Rational Rose para la modelación de los procesos de negocio.

Con la implementación de esta aplicación se obtiene una herramienta muy práctica para desarrollar las habilidades en la pronunciación inglesa de alumnos y profesores de la universidad de las ciencias informáticas.

## ÍNDICE DE CONTENIDO

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.</b> .....	<b>4</b>
1.1 INTRODUCCIÓN .....	4
1.2 JUEGOS PARA APRENDER FONÉTICA INGLESA EN EL MUNDO. ....	4
1.2.1 <i>Shephonics.</i> .....	4
1.2.2 <i>EnglishClub.Com.</i> .....	5
1.2.3 <i>Cambridge English Online.</i> .....	5
1.3 JUEGOS PARA APRENDER FONÉTICA EN LA UCI. ....	5
1.3.1 <i>Entorno Virtual de Aprendizaje (EVA)</i> .....	5
1.4 TECNOLOGÍAS Y LENGUAJES DE PROGRAMACIÓN.....	6
1.4.1 <i>PHP 5.2.</i> .....	6
1.4.2 <i>Sistema gestor de base de datos: PostgreSQL</i> .....	7
1.4.3 <i>AJAX.</i> .....	7
1.4.4 <i>JavaScript.</i> .....	8
1.4.5 <i>Framework CodeIgniter.</i> .....	9
1.5 HERRAMIENTAS UTILIZADAS. ....	11
1.5.1 <i>Rational Rose Enterprise para UML.</i> .....	11
1.5.2 <i>Photoshop.</i> .....	11
1.6. LENGUAJE DE MODELADO Y METODOLOGÍA DE DESARROLLO .....	11
1.6.1 <i>Lenguaje de modelado: UML.</i> .....	11
1.6.2 <i>Metodología de desarrollo: RUP.</i> .....	12
1.7 ARQUITECTURA CLIENTE/SERVIDOR.....	13
1.8 CONCLUSIONES .....	14
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA</b> .....	<b>16</b>
2.1 INTRODUCCIÓN .....	16
2.2 OBJETO DE ESTUDIO .....	16
2.2.1 <i>Problema y situación problemática.</i> .....	16
2.2.2 <i>Objeto de automatización.</i> .....	16
2.3 PROPUESTA DE SISTEMA. ....	17
2.3.1 <i>Modelo de Dominio.</i> .....	17
2.4 RELACIÓN DE LOS REQUERIMIENTOS. ....	19
2.4.1 <i>Listado de los requerimientos funcionales enumerados.</i> .....	19
2.4.2 <i>Listado de los requerimientos no funcionales:</i> .....	20
2.5 <i>Modelo de Casos de Uso del Sistema.</i> .....	22
2.5.1 <i>Diagrama de Casos de Uso del Sistema a Automatizar.</i> .....	23
2.5.2 <i>Patrones de casos de uso.</i> .....	23
2.5.3 <i>Descripción textual de los casos de uso. Ver Anexo 2.</i> .....	25
2.6 CONCLUSIONES. ....	25
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO.</b> .....	<b>26</b>
3.1 INTRODUCCIÓN. ....	26
3.2 ANÁLISIS. ....	26
3.2.1 <i>MODELO DE CLASES DE ANÁLISIS POR CASO DE USO.</i> .....	26

3.2.1 Diagramas de colaboración del análisis .....	27
3.3 DISEÑO.....	28
3.3.1 DIAGRAMA DE CLASES DEL DISEÑO.....	28
3.4 Modelo lógico de datos.....	29
3.4.2 Modelo físico de datos.....	30
1.6 PATRÓN DE ARQUITECTURA MODELO VISTA CONTROLADOR (MVC).....	31
3.5 CONCLUSIONES .....	31
<b>CAPÍTULO 4: IMPLEMENTACIÓN.....</b>	<b>32</b>
4.1 INTRODUCCIÓN.....	32
4.2 MODELO DE IMPLEMENTACIÓN.....	32
4.2.1 Diagrama de despliegue.....	32
4.2.2 Diagramas de componentes.....	33
CASO DE USO “SOLITARIO”.....	34
CASO DE USO “AUTENTICAR”.....	36
4.6 CONCLUSIONES .....	40
<b>CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD.....</b>	<b>41</b>
5.1 INTRODUCCIÓN.....	41
5.1 INTRODUCCIÓN.....	41
6.1 APLICAR MÉTODO DE ESTIMACIÓN PUNTOS POR CASOS DE USO.....	41
6.2 Cálculo de Puntos de Casos de Uso sin ajustar.....	41
6.4 Cálculo del Esfuerzo.....	46
6.5 Distribución del Esfuerzo entre las diferentes actividades de un proyecto:.....	47
7.2 Análisis de Costo.....	48
7.3 CONCLUSIONES.....	49
CONCLUSIONES .....	50
RECOMENDACIONES .....	51
REFERENCIAS BIBLIOGRÁFICAS .....	52
BIBLIOGRAFÍA.....	53
ANEXOS.....	55
Anexo 1: Diagramas de Colaboración.....	55
Anexo 2: Descripción textual de los casos de uso del sistema.....	63
GLOSARIO DE TÉRMINOS .....	77

**ÍNDICE DE TABLAS**

TABLA 1 ACTORES DEL SISTEMA.....	22
TABLA 2 FACTOR DE PESO DE LOS ACTORES SIN AJUSTAR.....	42
TABLA 3 ACTORES DEL SISTEMA PARA LA ESTIMACIÓN.....	42
TABLA 4 FACTOR DE PESO DE LOS CASOS DE USO SIN AJUSTAR.....	43
TABLA 5 CASOS DE USO DEL SISTEMA PARA LA ESTIMACIÓN.....	43
TABLA 6 FACTOR DE COMPLEJIDAD TÉCNICA.....	45
TABLA 7 FACTOR DE AMBIENTE.....	46
TABLA 8 ESFUERZO DEL PROYECTO.....	48
TABLA 9 DESCRIPCIÓN TEXTUAL CASO DE USO DE SISTEMA “SOLITARIO”.....	65
TABLA 10 DESCRIPCIÓN TEXTUAL CASO DE USO “CONCURSO”.....	67
TABLA 11 DESCRIPCIÓN TEXTUAL CASO DE USO “GESTIONAR INFORMACIÓN”.....	69
TABLA 12 DESCRIPCIÓN TEXTUAL CASO DE USO “GESTIONAR ROLES DE USUARIO”.....	71
TABLA 13 DESCRIPCIÓN TEXTUAL CASO DE USO “AUTENTICAR USUARIO”.....	72
TABLA 14 DESCRIPCIÓN TEXTUAL CASO DE USO “CONSULTAR INFORMACIÓN”.....	73
TABLA 15 DESCRIPCIÓN TEXTUAL CASO DE USO “GESTIONAR PUNTOS”.....	76
TABLA 16 DESCRIPCIÓN TEXTUAL CASO DE USO “CONSULTAR PUNTOS”.....	76

**ÍNDICE DE FIGURAS**

FIGURA 1 MODELO DE DOMINIO.....	18
FIGURA 2 CASOS DE USO DEL SISTEMA.....	23
FIGURA 3 DIAGRAMA DE CLASES DEL ANÁLISIS “VER NOTAS”.....	26
FIGURA 4 DIAGRAMA DE COLABORACIÓN CU “VER PUNTOS”.....	27
FIGURA 5 DIAGRAMA DE CLASES DEL DISEÑO WEB.....	28
FIGURA 6 DIAGRAMA DE CLASES PERSISTENTES.....	29
FIGURA 7 MODELO DE DATOS.....	30
FIGURA 8 DIAGRAMA DE DESPLIEGUE.....	32
FIGURA 9 DIAGRAMA DE COMPONENTES CU “VER NOTAS”.....	33
FIGURA 10 DIAGRAMA DE COMPONENTES CU “SOLITARIO”.....	34
FIGURA 11 DIAGRAMA DE COMPONENTES CU “CONCURSO”.....	35
FIGURA 12 DIAGRAMA DE COMPONENTES CU “AUTENTICAR”.....	36
FIGURA 13 DIAGRAMA DE COMPONENTES CU “GESTIONAR INFORMACIÓN”.....	37
FIGURA 14 DIAGRAMA DE COMPONENTES CU “GESTIONAR USUARIOS”.....	38
FIGURA 15 DIAGRAMA DE COMPONENTES CU “GESTIONAR PUNTOS”.....	39
FIGURA 16 DIAGRAMA DE COLABORACIÓN CU “VER NOTAS”.....	55
FIGURA 17 DIAGRAMA DE COLABORACIÓN CU “SOLITARIO”.....	55
FIGURA 18 DIAGRAMA DE COLABORACIÓN CU “CONCURSO”.....	56
FIGURA 19 DIAGRAMA DE COLABORACIÓN CU “AUTENTICAR”.....	57
FIGURA 20 DIAGRAMA DE CLASE DEL ANÁLISIS CU “CONSULTAR PUNTOS”.....	57
FIGURA 21 DIAGRAMA DE CLASE DEL ANÁLISIS CU “SOLITARIO”.....	58

FIGURA 22 DIAGRAMA DE CLASE DEL ANÁLISIS CU “GESTIONAR PUNTOS” .....	59
FIGURA 23 DIAGRAMA DE CLASE DEL ANÁLISIS CU “GESTIONAR INFORMACIÓN” .....	60
FIGURA 24 DIAGRAMA DE CLASE DEL ANÁLISIS CU “CONCURSO” .....	61
FIGURA 25 DIAGRAMA DE CLASE DEL ANÁLISIS CU “AUTENTICAR” .....	62

## INTRODUCCIÓN

La Universidad de las Ciencias Informáticas (UCI) cada vez adquiere más logros a nivel internacional con productos 100% propios. Mientras va ganando prestigio en un mercado nuevo y muy competitivo en el cual para que un producto sea aceptado independientemente del país o idioma de este, se debe emplear el inglés como lenguaje tanto para las especificaciones como para la interfaz del producto. El 95 por ciento de la información de Internet, la red de redes, la base de conocimiento más amplia hoy en día, está en inglés y los traductores automáticos no son suficientes para captar una idea a la perfección. Lo cual supone un nuevo desafío para los estudiantes de esta universidad: el dominio del inglés como herramienta imprescindible para nuestros productos. De las áreas del lenguaje, una de las de mayor dificultad para dominar es la pronunciación. Para lograr una comunicación exitosa es necesario desarrollar una pronunciación efectiva del idioma en cuestión. A partir del hecho de que el inglés tiene características distintivas en su pronunciación, los estudiantes suelen crear barreras que imposibilitan la adquisición del nuevo sistema fónico sobre todo si están en un medio estresante, esto hace que resulte difícil adquirir las habilidades necesarias para desarrollar un inglés oral lo suficientemente fluido y entendible, de ahí que surja la problemática de cómo motivar y propiciar la enseñanza-aprendizaje de la pronunciación del idioma inglés en nuestra universidad. A partir de un trabajo de diploma (referencia) que aborda este mismo tema y propone un juego didáctico que podría ayudar a solucionar esta situación presentamos este proyecto informático.

Dada la situación anteriormente planteada el **problema científico** radica en: ¿Cómo motivar y propiciar el aprendizaje de la fonética del idioma inglés en la Universidad de las Ciencias Informáticas?

El **objeto de estudio** de este trabajo de diploma es el proceso de enseñanza-aprendizaje de la pronunciación del idioma inglés en nuestra universidad. Teniendo como **campo de acción** el estudio de la pronunciación inglesa en la UCI.

A partir del problema anteriormente descrito se ha definido como **objetivo general** de esta tesis el desarrollar un juego didáctico para la enseñanza – aprendizaje de la pronunciación en inglés.

Para dar cumplimiento a este objetivo general se definieron las siguientes **tareas de investigación concretas**:

- Realización de entrevistas al cliente.
- Investigación acerca de las plataformas a utilizar para la implementación del Sistema.
- Revisión de posibles propuestas realizadas anteriormente como base para el desarrollo del Sistema.
- Realización de las actividades correspondientes al análisis y diseño del Sistema.
- Desarrollo de la base de datos del Sistema.
- Implementación del Sistema.

Para la realización y cumplimiento de los objetivos y las tareas propuestas se utilizó el siguiente método teórico:

**Analítico sintético**: Son dos procesos inherentes al pensamiento, operaciones lógicas importantes; que nos permiten; como métodos teóricos, buscar la esencia de los fenómenos, los rasgos que los caracterizan y los distinguen.

Su objetivo en una investigación es analizar las teorías, documentos, etc.; permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.

El documento estará estructurado en cinco capítulos y las correspondientes conclusiones, recomendaciones, referencias bibliográficas, bibliografía, anexos y glosario de términos.

**Capítulo 1. Fundamentación Teórica**: Se describen las tendencias actuales de las tecnologías, se muestran ejemplos de aplicaciones existentes y se hace un análisis de las herramientas seleccionadas para la implementación del sistema propuesto, analizando sus características, ventajas y desventajas.

**Capítulo 2.** Características del Sistema: Se identifican los problemas existentes y las necesidades de los usuarios y se hace una descripción de cómo se realiza el proceso objeto de estudio de nuestra investigación. Se exponen los artefactos que permiten comprender la estructura y la dinámica del proceso abordado según la metodología utilizada. Se realiza el levantamiento de los requerimientos funcionales y no funcionales. Se hace el modelo de dominio en el cual quedan representadas las clases que componen el negocio. Se representan los diferentes casos de uso del sistema con empleando patrones de casos de uso.

**Capítulo 3.** Análisis y Diseño del Sistema: Se exponen los diagramas de clases del análisis y el diseño, el modelo de datos y las clases persistentes, los diagramas de secuencias, el diagrama Entidad Relación de la BD (utilizando para su modelado el Lenguaje Unificado de Modelación(UML)), la descripción de las tablas y se explican de manera general las clases utilizadas. Dejando plasmado además el patrón de diseño de arquitectura a utilizar.

**Capítulo 4.** Implementación del sistema: Se desarrollan los artefactos del flujo de trabajo Implementación, entre ellos los diagramas de componentes y modelos de despliegue del modelo de implementación.

**Capítulo 5.** Estudio de factibilidad: Se realizará un estudio de factibilidad mediante el Método de Estimación por Puntos de Casos de Uso para de esta manera determinar si es factible o no la implementación de este proyecto.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.**

### **1.1 Introducción**

En este capítulo se realizará un estudio sobre las diferentes aplicaciones Web existentes en el mundo y en la universidad que incluyan juegos para la enseñanza de la fonética inglesa. Se definirán además las herramientas, metodologías, arquitectura y tecnologías que se utilizarán para desarrollar la aplicación.

### **1.2 Juegos para aprender fonética inglesa en el mundo.**

En el mundo existen una gran variedad de aplicaciones Web que incluyen juegos para aprender fonética inglesa. La mayoría desarrolladas por instituciones dedicadas a la enseñanza del idioma inglés como es el caso de la prestigiosa universidad de Cambridge. Pero estas aplicaciones a pesar de su utilidad solo pueden ser usadas a través de internet, lo cual dificulta que puedan ser empleadas por nuestros profesores en sus clases, ya que esto implicaría un gasto considerable en nuestra cuenta de acceso a internet. Por lo tanto no serían soluciones prácticas a aplicar dentro de la universidad.

Existen muchas aplicaciones de escritorio con juegos para aprender fonética inglesa que incluyen además sonidos como forma de retroalimentación. Estas aplicaciones con de hecho muy efectivas y usadas en el mundo, pero comparadas con las aplicaciones Web presentan ciertas desventajas: tendrían que ser instaladas en cada computadora de la universidad, no permite a los profesores llevar un control del progreso de los estudiantes, no permiten en su gran mayoría la interacción entre usuarios y gran parte de estas aplicaciones son propietarias. Por lo tanto tampoco serían soluciones prácticas a aplicar dentro de la universidad.

#### **1.2.1 Shephonics.**

Sephonics es una aplicación de tipo escritorio para aprender y practicar el alfabeto fonético inglés a través de diversos ejercicios y juegos.

Sephonics incluye siete tipos de ejercicios para practicar la pronunciación inglesa y el alfabeto fonético, como ejercicios de emparejar un sonido a una muestra fonética, transcribir del texto fonético al texto ordinario, discriminación de sonidos, deletreo y un juego fonético de memoria.

### 1.2.2 EnglishClub.Com.

Este sitio web nombrado EnglishClub.com apoya el uso de juegos para el aprendizaje del inglés, fácilmente accesible y disponible en internet .Contiene además un amplio repertorio de opciones didácticas que incluyen: El Ahorcado (Hangman), crucigramas (Crosswords), Chistes (jokes), Palabras Desordenadas (Jumbled Words), etc. (1)

### 1.2.3 Cambridge English Online.

Dado el ingente número de Webs existentes sobre idiomas, las instituciones más prestigiosas en materia educativa del Reino Unido no podían quedar al margen. Por ello, la **Cambridge University Press** ha lanzado, en colaboración con la **BBC, The British Council** y otros organismos, una página dotada con los más innovadores recursos de aprendizaje de idiomas: **Cambridge Online.**(7)

Esta web, **especializada en materia de fonética**, aunque contiene medios de aprendizaje de otras áreas, presenta **canciones, juegos, tests, ejercicios, lecturas** y otras alternativas para mejorar nuestra pronunciación de la lengua de **Shakespeare**.

Como vemos, se trata de **juegos divertidos** que, a la vez nos obligan a aprender el idioma. Pero, quizá el mejor de todos sea el llamado '**shoot a symbol**'(**Dispara a un símbolo**), que consiste en una especie de tiro. A partir de una palabra dada, van apareciendo fonemas vocálicos, consonánticos y diptongos en la pantalla, y debemos disparar al fonema que corresponde a esa palabra por su orden correspondiente. (2)

## 1.3 Juegos para aprender fonética en la UCI.

### 1.3.1 Entorno Virtual de Aprendizaje (EVA)

Este es un espacio de apoyo al proceso de formación de la Carrera de Ingeniería en Ciencias Informáticas y a los programas de postgrado que tributan a la formación del claustro de profesores de La Universidad.

Es un espacio virtual donde los profesores pueden implementar estrategias de enseñanza–aprendizaje complementarias a las clases presenciales, así como diseñar cursos semipresenciales o totalmente a distancia. Los estudiantes disponen de un poderoso medio en el cual pueden obtener, utilizar o compartir materiales didácticos.

Desafortunadamente la parte del EVA correspondiente a la asignatura inglés no incluye aún herramientas para la enseñanza de la fonética. Esto se debe primeramente a que no está incluida en el programa de la asignatura además de la dificultad que presenta este tema lo cual hace necesario el uso de nuevos métodos que sean más motivadores para los estudiantes.

#### **1.4 Tecnologías y lenguajes de Programación.**

##### **1.4.1 PHP 5.2.**

Este lenguaje tiene muchas características para su fácil manejo y aprendizaje ya que utiliza instrucciones sencillas que cualquier programador promedio puede entender. " 1.PHP es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor". Además tiene funciones incorporadas para manejadores de bases de datos como MySQL, PostgreSQL y hasta Oracle. Pero probablemente lo que lo hace superar a otros lenguajes es el hecho de que está publicado bajo la PHP License, esta es una licencia de código abierto, lo cual básicamente significa que esta licencia no debe impedir la venta o entrega de programas que lo utilicen, además esta licencia no demanda ningún impuesto de dicha venta, la Free Software Foundation considera esta licencia como software libre. Esta licencia está disponible en [www.php.net/license](http://www.php.net/license).

Otras características no menos importantes son:

- No requiere definición de tipos de variables.
- Posee tratamiento de errores.
- Es un lenguaje multiplataforma.
- Es un producto de código abierto, soportado por una gran comunidad de desarrolladores que se encargan de encontrar y reparar los fallos de funcionamiento. (3)

### 1.4.2 Sistema gestor de base de datos: PostgreSQL.

PostgreSQL es un gestor de bases de datos relacional orientada a objetos, libre bajo licencia. Presenta muchas ventajas que lo convierten en un gestor fácil de usar con una interfaz agradable y sencilla.

Propiedades que presenta:

- **Atomicidad:** Asegura la realización de una operación, por lo que ante un fallo del sistema esta no queda a medias.
- **Consistencia:** Posibilita la ejecución de aquellas operaciones que no van a romper las reglas y directrices de integridad de la base de datos.
- **Aislamiento:** Mediante un sistema denominado MVCC (Acceso concurrente multiversión) asegura que una operación no pueda afectar a otras, de esta manera dos transacciones sobre la misma información no genera error.
- **Durabilidad:** Asegura la permanencia de una operación realizada, y aunque falle el sistema esta no podrá deshacerse.

PostgreSQL presenta las siguientes características:

- Se puede instalar un número ilimitado de veces sin temor de sobrepasar la cantidad de licencias.
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos.
- Posee confiabilidad, seguridad y flexibilidad.

### 1.4.3 AJAX.

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), facilita la creación de aplicaciones interactivas en la Web que se ejecutan en el navegador de los usuarios y mantienen comunicación asíncrona con el servidor; posibilitando realizar cambios sobre una página sin necesidad de recargarla, aumentando de esta forma la interactividad, velocidad y usabilidad de la misma.

“AJAX no es una tecnología en sí mismo. En realidad, se trata de la unión de varias tecnologías que se desarrollan de forma autónoma y que se unen para lograr nuevas funcionalidades.”(8)

AJAX está conformado por:

- XHTML y CSS, para crear una presentación basada en estándares.

- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

Además,

- Provee un mecanismo para mezclar y hacer coincidir XML con XHTML.
- Las aplicaciones son más rápidas e interactivas, al estilo aplicaciones de escritorio.
- Reduce de manera significativa tener que cargar información continuamente del servidor, ya que actualiza porciones de la página en vez de la página completa.
- Cuando se utiliza AJAX adecuadamente en el desarrollo de una aplicación, se reduce de manera significativa los tiempos de carga inicial. (4)

#### 1.4.4 JavaScript.

”JavaScript es un simple lenguaje de programación a medio camino entre el Java más complejo y el HTML más intuitivo, del cual, para algunos observadores, no sería sino una extensión.” (5)

Javascript es un lenguaje interpretado ya que son los navegadores los que lo interpretan y por tanto ejecutan los programas escritos en este lenguaje. De esta forma podemos enviar documentos a través de la Web que llevan incorporados el código fuente de programas, convirtiéndose de esta forma en documentos dinámicos y dejando de ser simples fuentes de información estáticas.

Las dos principales características de JavaScript son, por un lado que es un lenguaje basado en objetos (es decir, el paradigma de programación es básicamente el de la programación dirigida a objetos, pero con menos restricciones), y por otro que es orientado a eventos. Esto implica que gran parte de la programación en JavaScript se centra en describir objetos y escribir funciones que respondan a movimientos del mouse, pulsación de teclas, apertura y cerrado de ventanas o carga de una página, entre otros eventos. (9)

Entre sus inconvenientes se pueden mencionar primero el hecho de que en ocasiones se debe modificar el mismo código para que se adapte a los diferentes navegadores. Otro inconveniente es el

hecho de que este lenguaje debe estar activado en los navegadores para que su uso sea posible en estos.

#### 1.4.5 Framework CodeIgniter

El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas de software. Se pueden encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, etc. En general, el término framework, se refiere a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación.

Existen varios tipos de frameworks Web: orientados a la interfaz de usuario, como Java Server Faces, orientados a aplicaciones de publicación de documentos, como Cocoon, orientados a la parte de control de eventos, como Struts y algunos que incluyen varios elementos como Tapestry. La mayoría de los frameworks Web se encargan de ofrecer una capa de controladores de acuerdo con el patrón Modelo Vista Controlador (MVC) o con el modelo 2 de Servlets y JSP, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación.

**CodeIgniter** es un entorno de trabajo web para el desarrollo de aplicaciones web en PHP que facilita la escritura de código fuente en este lenguaje y reduce el tiempo necesario para lograrlo. Está basado en el sistema modelo vista controlador.

Una de las características más interesantes de **CodeIgniter** es el elevado número de clases que incluye para trabajar con distintos objetos: calendario, bases de datos, correo electrónico, manipulación de imágenes, lenguaje, tablas, sesiones, etcétera. Además se encuentra bajo una licencia open source Apache/BSD-style,

A diferencia de otros frameworks, **CodeIgniter** cuenta con una documentación excelente que permite conocer todos los secretos de este entorno de trabajo. (22)

Para poder ejecutar **CodeIgniter** se debe contar con un servidor web como Apache y el lenguaje de programación PHP, que fueron precisamente los seleccionados para desarrollar la aplicación. También es posible su utilización de forma remota. A continuación se muestran algunas de las razones por las que se escoge este framework.

- ✓ *Rendimiento.* **CodeIgniter** tiene muy buen performance en comparación con Zend o CakePHP.
- ✓ *Documentación.* Tiene una muy buena documentación en línea, además de una comunidad de usuarios muy activos (hoy en día) que permite hacerle mejoras y detectar errores rápidamente.
- ✓ *Estilo arquitectónico.* El uso de **CodeIgniter** se basa en el patrón Modelo-Vista-Controlador (MVC), separando la vista, del acceso a los datos y del controlador.
- ✓ *Facilidad de uso.* La curva de aprendizaje es sumamente corta y se puede aprender con facilidad.
- ✓ *Active Record.* El manejo y abstracción de base de datos es sencillo pero muy efectivo, e implementa el patrón Active Record que se utiliza para acceder a la base de datos usando consultas predefinidas.
- ✓ *Estructura organizada.* La estructura de archivos es muy clara y ofrece gran flexibilidad en caso que necesite cambiarse por algún motivo. La división en modelos, vistas, controladores, librerías, ayuda, configuración, etc. es muy práctica y comprensible.
- ✓ *Ruteo.* Los mecanismos de ruteo de CodeIgniter favorecen a la optimización para buscadores y otras aplicaciones.

**CodeIgniter** es un poderoso Framework para PHP que facilita la escritura de código repetitivo, y a comparación de otros Frameworks como CakePHP, Symfony o Zend Framework, **CodeIgniter** es más rápido y es totalmente extensible y altamente compatible con gran variedad de versiones y configuraciones de PHP tales como la 4 y la 5.

## **1.5 Herramientas Utilizadas.**

### **1.5.1 Rational Rose Enterprise para UML.**

Rational Rose al igual que los demás productos de la familia Rational Rose incluye soporte para Lenguaje Unificado de Modelado (UML), es una herramienta de diseño de software destinado para el modelado visual y componentes de construcción de aplicaciones de software a nivel de empresa. Rational Rose Enterprise además es multiplataforma a diferencia de otras herramientas similares. Es una potente herramienta para el desarrollo web. Su principal desventaja es la necesidad de alta capacidad de procesamiento.

Rational Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros del equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de Software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común. Otra ventaja es que los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

### **1.5.2 Photoshop.**

Adobe Photoshop es una aplicación informática de edición y retoque de imágenes. Actualmente es un estándar mundial para retoque fotográfico, diseño, fotografía, diseño web, composición fotográfica, pintura digital y video, entre otros. Para lograr estas técnicas Photoshop posee grandes ventajas en su uso como la utilización de capas y una versátil caja de herramientas. Las principales desventajas que presenta son aprender a dominar el producto y que es un software propietario bastante caro.

## **1.6. Lenguaje de modelado y metodología de desarrollo.**

### **1.6.1 Lenguaje de modelado: UML.**

UML (Unified Modeling Language): Es un lenguaje utilizado para el modelado de un sistema, permitiendo representar todas las fases de un proyecto informático, desde el análisis con los casos de uso, hasta la implementación y configuración con los diagramas de despliegue.

UML permite:

- Visualizar gráficamente un sistema de manera clara y precisa.
- Especificar cuáles son las características de un sistema antes de su construcción.
- Construir sistemas diseñados a partir de modelos especificados.
- Documentar los elementos gráficos del sistema desarrollado para futuras revisiones.

UML aporta las siguientes ventajas:

- Permite realizar una verificación y validación del modelo realizado.
- El modelado con UML es independiente del lenguaje de implementación, de tal forma que los diseños realizados se pueden implementar en cualquier lenguaje.
- Permite generar código a partir de los modelos y a la inversa, lo que posibilita la constante actualización.
- Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es el suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo en una empresa y diseño de la estructura de una organización.

**1.6.2 Metodología de desarrollo: RUP.**

“Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. El Proceso Unificado es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipo de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.” (6)

RUP (Rational Unified Process) es un proceso de desarrollo de software que junto al Lenguaje Unificado de Modelado UML, constituye una metodología estándar utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

- Se utiliza en proyectos que se desarrollan a largo plazo.
- Permite una mejor comunicación entre ingenieros de software y de negocio, ya que maneja un lenguaje común, facilita la comprensión de procesos del negocio y su traducción a las funcionalidades que brindará el sistema.
- Genera un volumen considerable de documentación, posibilitando que los cambios realizados en los miembros del equipo no resulte un factor negativo para el avance del proyecto.
- Propone el desarrollo en ciclos e iteraciones con los respectivos artefactos que se generan, siendo esto un elemento importante para alcanzar una categoría de certificación en el desarrollo del software.
- Asegura la producción de software de calidad dentro de plazos y presupuestos predecibles.

En conclusión, los verdaderos aspectos definitorios del Proceso Unificado se resumen en tres aspectos claves:

- Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de este momento los casos de uso guían el proceso de desarrollo, ya que los modelos que se obtienen en los diferentes flujos de trabajo, representan la realización de los casos de uso.
- Centrado en la arquitectura: La arquitectura muestra la visión común del sistema en la que el equipo del proyecto y los usuarios deben estar de acuerdo. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura.
- Iterativo e incremental: Propone que cada fase se desarrolle en iteraciones que involucran actividades de todos los flujos de trabajo. Es práctico dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

### **1.7 Arquitectura Cliente/Servidor.**

La arquitectura cliente-servidor es una nueva tendencia en el desarrollo de redes locales, que tiene como objetivo optimizar el uso tanto del hardware como del software a través de la separación de

funciones: el cliente que maneja la porción de la aplicación y el servidor que administra los procesos de almacenamiento y recuperación de los datos.

Puede presentarse como uno a varios clientes y uno o más servidores, junto con un sistema operativo y una plataforma de comunicación para formar un sistema cooperativo que permita la computación distribuida, el análisis y la presentación de datos.

Un único servidor típicamente sirve a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente.

#### Características de la arquitectura Cliente/Servidor:

- El servidor presenta una interfaz única y bien definida a todos sus clientes.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor no afectan al cliente.

En nuestra propuesta esta arquitectura se aplica para que el cliente tenga lo referente a la interfaz del juego así como otras opciones visuales igualmente sencillas en su navegador. Mientras que las funciones estarán ejecutándose en el servidor. Teniendo en cuenta que un servidor da servicio a múltiples clientes de forma concurrente, utilizando esta arquitectura será más fácil desarrollar el juego online. El servidor será el puente para la información entre los usuarios que estén jugando.

### **1.8 Conclusiones**

Teniendo en cuenta los puntos analizados en este capítulo se evidencia que a pesar de existir muchos y muy efectivos programas en el mundo para el aprendizaje de la fonética inglesa, esta propuesta es la mejor solución a aplicar en nuestra universidad. Las demás propuestas internacionales al ser aplicaciones Web implican un gasto considerable de la cuenta de los estudiantes y profesores si son visitadas frecuentemente. Actualmente no existe aplicación alguna dentro de la UCI para este fin.

La elección de las herramientas y tecnologías para esta propuesta fue satisfactoria pues las mismas presentan características que permitirán desarrollarla abarcando todas sus funcionalidades. Por lo que

quedan definidos los lenguajes, herramientas y tecnologías anteriormente mencionadas como los que serán utilizados en el desarrollo de esta aplicación.

## **CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA**

### **2.1 Introducción**

En este capítulo se lleva a cabo un análisis del negocio que corresponde a la situación problemática planteada y de la necesidad de una aplicación web para darle solución al mismo. Se explicarán las características de los procesos de negocio actual y se especificarán el objeto de automatización y la propuesta del sistema.

### **2.2 Objeto de estudio**

#### **2.2.1 Problema y situación problemática.**

En nuestra universidad el dominio del idioma Inglés es necesario por las características de la carrera. Pero los estudiantes en general presentan dificultades en el aprendizaje del mismo. Sobre todo en lo que respecta a la comunicación oral, los estudiantes en su mayoría no desarrollan una pronunciación lo suficientemente buena. Una posible solución a este problema sería el uso del lenguaje fonético para ayudar a los estudiantes a mejorar su pronunciación.

Actualmente en el mundo existen muchas herramientas para la enseñanza-aprendizaje de la fonética inglesa pero la mayoría de estas aplicaciones o son muy caras o son sitios web publicados en internet que consumirían gran parte de nuestra cuenta mensual de acceso a internet para poderlos usar el tiempo necesario.

La necesidad de un software propio de la Universidad que esté disponible para tantos estudiantes como sea necesario para la práctica de la fonética inglesa. Que sirva de material de apoyo a los profesores de la asignatura y que además incentive a nuestros estudiantes conforman las bases de nuestra situación problemática.

#### **2.2.2 Objeto de automatización.**

Las clases de fonética que se imparten actualmente en su mayoría dependen solo de las herramientas que utilice el profesor y estas no siempre permiten la interacción de todos los estudiantes de la clase. Además la fonética puede ser un tema complejo para estudiantes que no tengan un nivel muy avanzado

en inglés. La implementación de nuestro software tiene como objetivo desarrollar un juego fonético en plataforma web que ayude a los estudiantes y profesores de la asignatura inglés a aprender o mejorar sus habilidades fonéticas de una manera fácil y motivacional.

### **2.3 Propuesta de sistema.**

La aplicación que se desea implementar sería un juego de tablero que dada una palabra en inglés el jugador (ya sea estudiante o profesor) deberá transcribirla a lenguaje fonético utilizando fichas con los 51 símbolos existentes, en el tablero. Aunque sitios como Cambridge English Online y English club presentan también juegos online para practicar la fonética ninguno utiliza un juego de este tipo. Además nuestro juego incluye la novedad de permitir que el juego sea multijugador o sea dos jugadores serán rivales en este juego de manera que este tipo de variante ayudará a medir la habilidad de algunos estudiantes frente a otros. La aplicación estará disponible para toda la universidad y solo se necesitaría la cuenta UCI para acceder a esta. Se almacenarán los puntos acumulados por aquellos que jueguen para tener una idea de su avance en este tópico.

#### **2.3.1 Modelo de Dominio.**

Debido a que en nuestro proyecto los procesos no están del todo definidos se decidió emplear un modelo de dominio en vez de uno de negocio completo. Este modelo de dominio muestra las clases involucradas en el negocio así como las relaciones entre ellas, además se explica en qué consiste y su importancia.

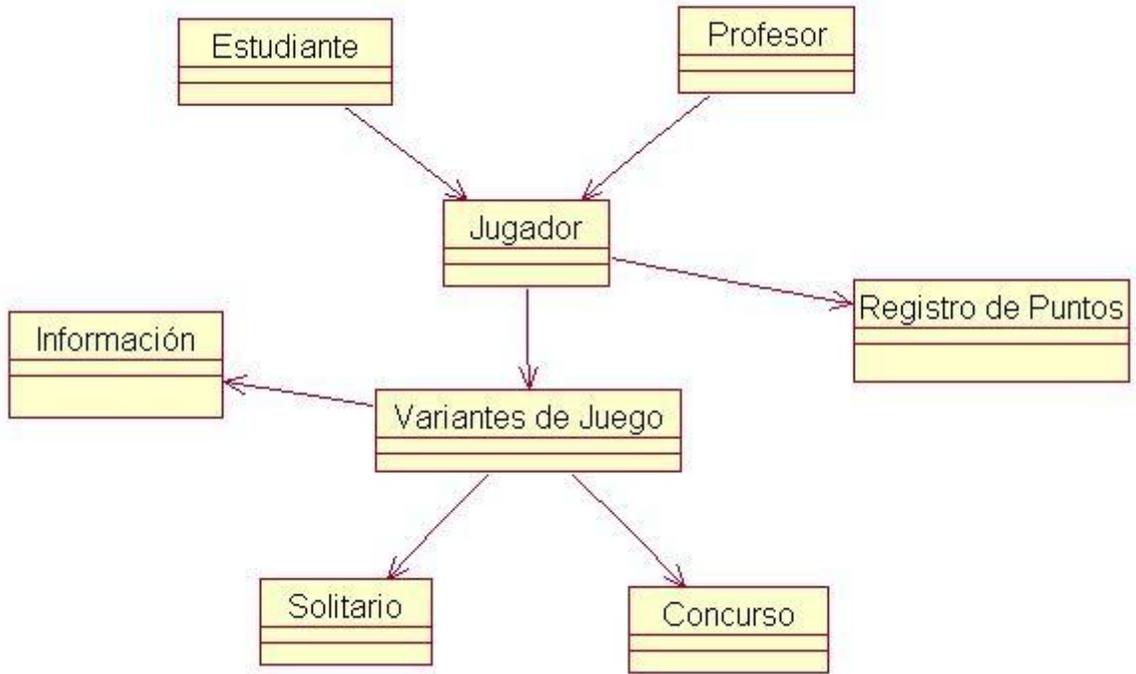


Figura 1 Modelo de dominio

**Concurso:**

Objetivo del concurso: oponer a dos jugadores transcribiendo palabras en el panel para ver cuál de ellos obtiene más puntos. Cada jugador desarrollará 3 jugadas independientes. El jugador que obtenga más puntos será el ganador.

**Solitario:**

En esta variante un solo jugador transcribirá palabras que irán aumentando su complejidad en tres niveles con tres oportunidades cada uno.

Registro de puntos:

Contiene organizado por facultades y grupos los puntos acumulados de los jugadores.

**Información:**

Contiene la información que usarán los juegos, las palabras en inglés y sus transcripciones fonéticas.

**Estudiante:**

Los estudiantes de la universidad.

**Profesor:**

Los profesores de la universidad.

**Jugador:**

Es quién utiliza el juego.

**Registro de puntos:**

Los puntos que acumula el jugador.

**2.4 Relación de los requerimientos.**

**2.4.1 Listado de los requerimientos funcionales enumerados.**

1. Autenticar usuarios.
2. Controlar la información de los juegos.
  - 2.1 Agregar información a los juegos.
  - 2.2 Modificar la información de los juegos.
  - 2.3 Eliminar información de los juegos.
3. Controlar el registro de puntos.
  - 3.1 Adicionar puntos del estudiante.
  - 3.2 Modificar los puntos del estudiante.
  - 3.3 Eliminar los puntos del estudiante.
4. Mostrar variantes de juego.
5. Mostrar listado de posibles jugadores.
6. Controlar las oportunidades y los niveles de cada juego.

## 2.4.2 Listado de los requerimientos no funcionales:

### **De seguridad:**

La información será almacenada en bases de datos, dejando registro de todas las operaciones realizadas.

El uso y manejo del sistema estará controlado ya que toda la información podrá ser consultada solamente por el personal autorizado.

### **De confiabilidad:**

Todas las salidas del sistema tienen que ser precisas y veraces.

Toda la información debe estar protegida del acceso no autorizado. Solo el personal acreditado podrá administrar la información solicitada.

El sistema tendrá la capacidad de recuperarse rápidamente ya que se le harán salvaguardas de seguridad como medida preventiva en caso de cualquier fallo.

### **De disponibilidad:**

El tiempo medio de reparación debe ser menor de 1 día.

Los usuarios del sistema deben tener acceso (según sus permisos) en todo momento a la información solicitada.

### **De rendimiento:**

El sistema con el uso de las tecnologías Web debe tener un tiempo de respuesta rápido y eficiente, inferior a 5 segundos.

Después de instalado el software, el mismo debe ser capaz de soportar gran cantidad de usuarios conectados simultáneamente, teniendo en cuenta de que en la universidad actualmente hay alrededor de 10 000 estudiantes sería lógico el pensar en esta cifra como posible cantidad máxima de usuarios.

**De portabilidad:**

Este producto podrá ser utilizado en cualquier sistema operativo, o sea, será multiplataforma pues en su implementación se utilizarán herramientas multiplataforma.

**De apariencia o interfaz externa:**

La interfaz en su totalidad debe ser amigable, sugerente, intuitiva e interactiva para que pueda captar la atención del usuario.

Debe contener un diseño sencillo, con pocas imágenes y gráficos para acelerar la velocidad de respuesta del sistema.

La aplicación correrá a pantalla completa para que se pueda visualizar correctamente.

La interfaz debe presentar solamente las funcionalidades del rol que esté utilizando el Sistema, para lograr la concentración del usuario en las actividades que esté realizando.

**De hardware:**

Se necesita una PC que posea un procesador Pentium II o superior y 256 megabytes (MB) de memoria RAM como mínimo. El servidor Web, al igual que el servidor de base de datos, debe tener 512 MB de RAM y 60 GB de disco duro mínimo, todas las máquinas implicadas en la funcionalidad de la aplicación deben estar conectadas a la red al menos a 100 Mbps de velocidad.

**De software:**

- Sistema Operativo Linux, Windows 95 ó superior.
- Servidor Web Apache.
- Gestor de Base de Datos PostgreSQL 8.2.
- Framework de desarrollo: CodeIgniter.

**De usabilidad:**

La interfaz será fácil de usar por los diversos usuarios que interactúen con ella.

**De soporte:**

El producto será probado, instalado y configurado por el cliente y un administrador que se encargará también de darle mantenimiento.

Se deben ofrecer servicios de adiestramiento al personal que va a trabajar con el software.

**Legales:**

El software así como su documentación son propios de la UCI.

**De ayuda y documentación en línea:**

El sistema proporcionará en todo momento la documentación necesaria para que los usuarios puedan acceder a la misma en caso de que necesiten realizar alguna consulta.

**2.5 Modelo de Casos de Uso del Sistema.**

Actores	Justificación
Usuario	Se autentica en el sistema utiliza los servicios disponibles de acuerdo a su rol.
Jugador	Puede seleccionar cualquiera de las tres variantes de juegos disponibles (concurso, competición, solitario).
Estudiante	Interactúa con la aplicación durante su proceso de aprendizaje. Es un jugador, puede consultar el diccionario o utilizar algunos de los otros servicios disponibles de la aplicación.
Profesor	También es un jugador pero además monitorea las actividades de sus estudiantes, gestiona el contenido para asegurarse que sea el apropiado.
Administrador del sistema	Gestiona los usuarios.

**Tabla 1 Actores del sistema.**

### 2.5.1 Diagrama de Casos de Uso del Sistema a Automatizar.

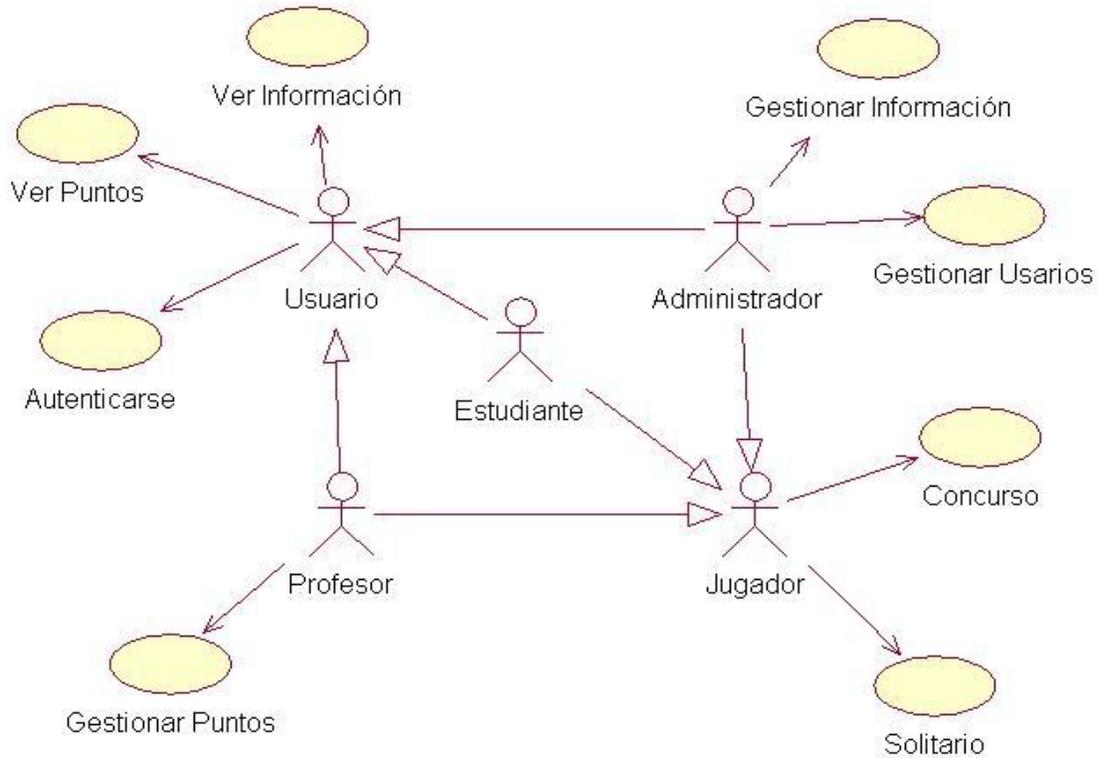


Figura 2 Casos de Uso del sistema

### 2.5.2 Patrones de casos de uso.

Los patrones son un grupo de clases/objetos colaborando entre sí que se pueden abstraer de un conjunto de escenarios general. Los patrones son un medio excelente para lograr reutilización y desarrollo robusto. A medida que los patrones se descubren en todo nuevo proyecto, se puede reutilizar la plantilla básica del patrón desde modelos previos con los nombres de las variables apropiadas modificados para el proyecto en curso.

Para desarrollar este modelo de casos de uso se aplicaron patrones que permitieron representar de una forma más precisa los requisitos reales para alcanzar mejores resultados, lo cual facilita considerablemente el desarrollo de este trabajo.

*CRUD* (Crear, Obtener, Actualizar, Eliminar): Este patrón se usó para modelar todas las operaciones que se pueden realizar sobre una parte de información de un tipo determinado; ya sea crearla, leerla, actualizarla y eliminarla. Es importante destacar que este patrón se usa cuando todos los flujos contribuyen al mismo valor de negocio y son cortos y simples.

*Múltiples actores (Roles comunes)*: Este patrón se utilizó para representar actores que interpretan el mismo papel en determinado caso de uso. Este rol es representado por otro actor, que contiene de forma hereditaria los actores que comparten este rol.

Estos patrones permitieron una modelación del sistema más precisa, además de que ayudaron a resolver problemas durante la modelación del sistema. Obteniendo modelos de mayor calidad de una forma más rápida.

### **2.5.3 Descripción textual de los casos de uso. Ver Anexo 2.**

### **2.6 Conclusiones.**

En este capítulo se describieron los procesos y actividades que se realizan en el negocio planteado, se describieron las funcionalidades del sistema a través de los casos de uso así como las propiedades y cualidades que el sistema debe tener. Al culminar este capítulo quedan definidas las características del sistema a implementar.

## CAPÍTULO 3: Análisis y diseño.

### 3.1 Introducción.

En el presente capítulo se analizarán los requerimientos definidos durante la caracterización del sistema. Mediante la realización de los diagramas de análisis y diseño, se modelará el sistema de modo que estos reflejen su estructura interna.

### 3.2 Análisis.

El análisis se realiza durante la fase de elaboración y consiste en obtener una visión del sistema que se preocupa de ver QUÉ hace.

#### 3.2.1 Modelo de clases de análisis por caso de uso.

Aquí se refleja a través de los diferentes diagramas del análisis como están distribuidos los diferentes objetos que componen la aplicación y cómo funciona la comunicación entre ellos.



Figura 3 Diagrama de Clases del análisis “Ver Notas”.

### 3.2.1 Diagramas de colaboración del análisis.

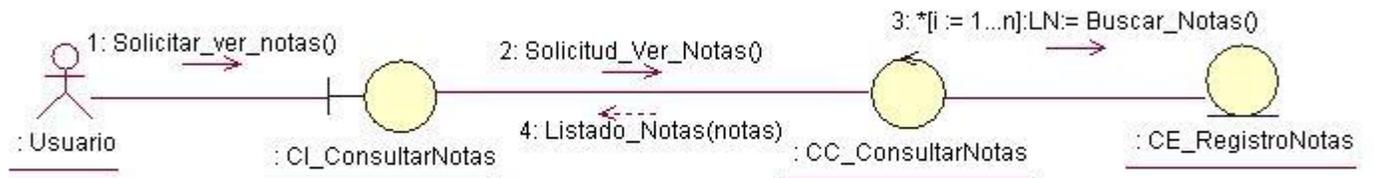


Figura 4 Diagrama de colaboración CU “Ver Puntos”.

### 3.3 Diseño.

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, o sea, CÓMO cumple el sistema sus objetivos.

#### 3.3.1 Diagrama de clases del diseño.

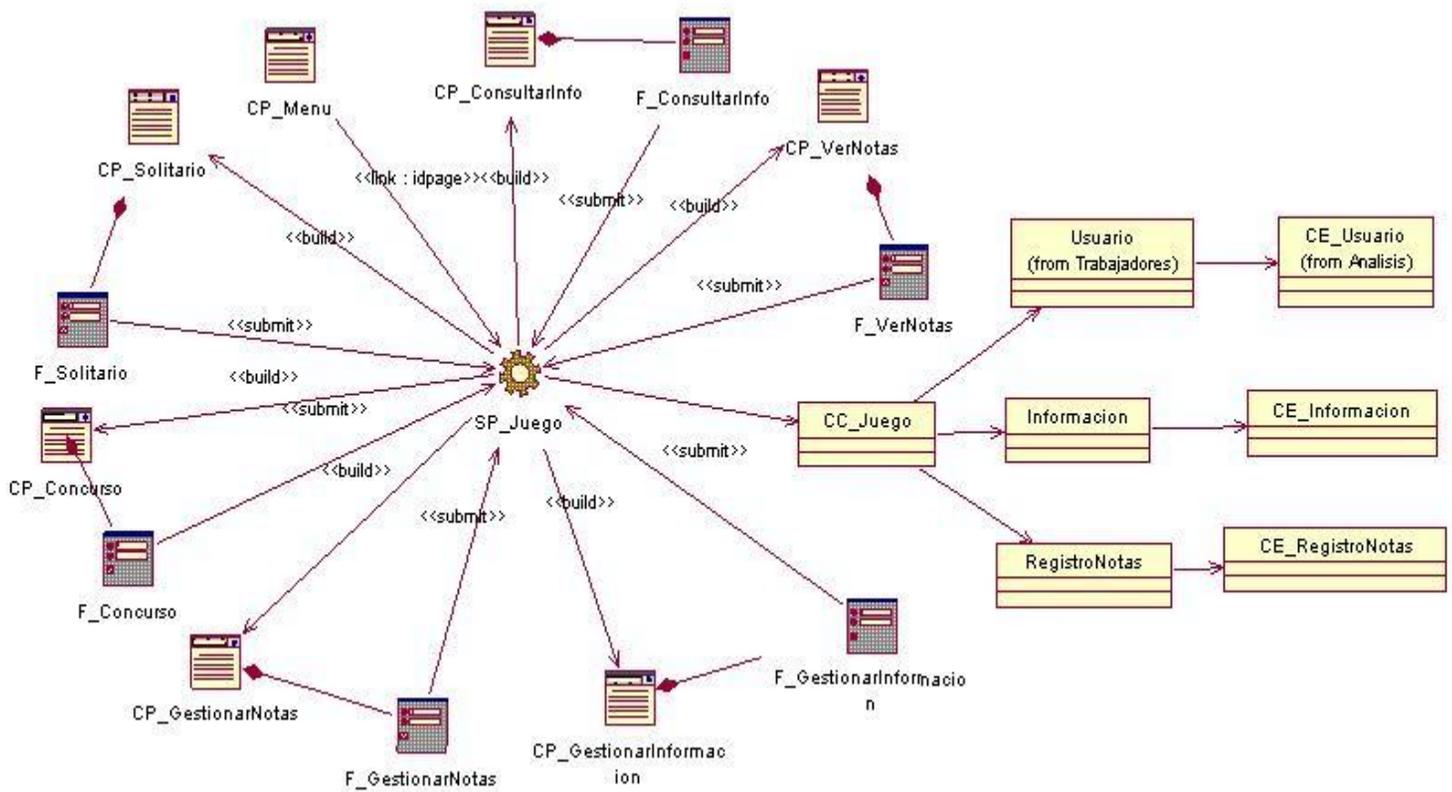


Figura 5 Diagrama de clases del diseño Web.

### 3.4 Modelo lógico de datos.

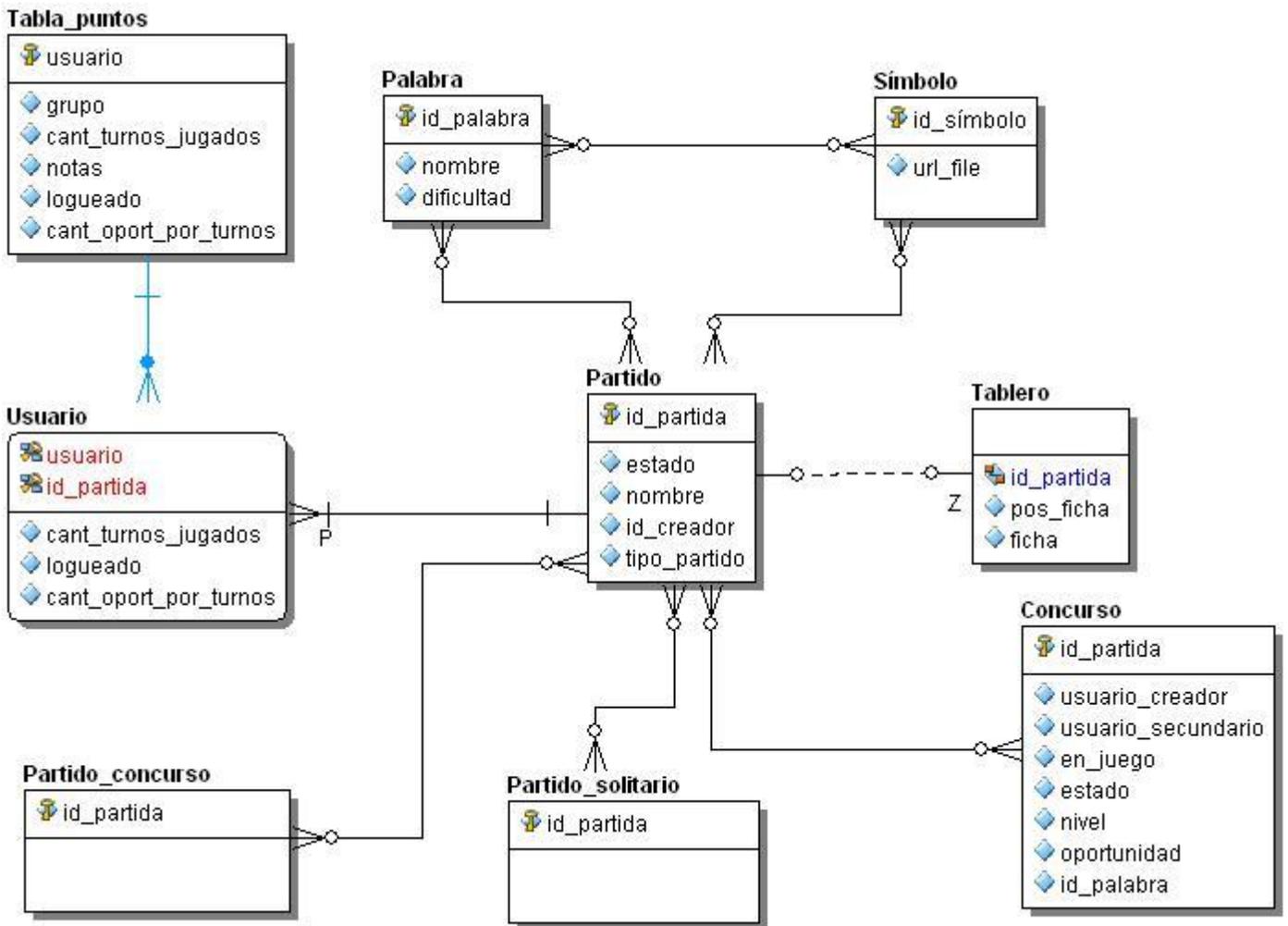


Figura 6 Diagrama de clases persistentes.

3.4.2 Modelo físico de datos.

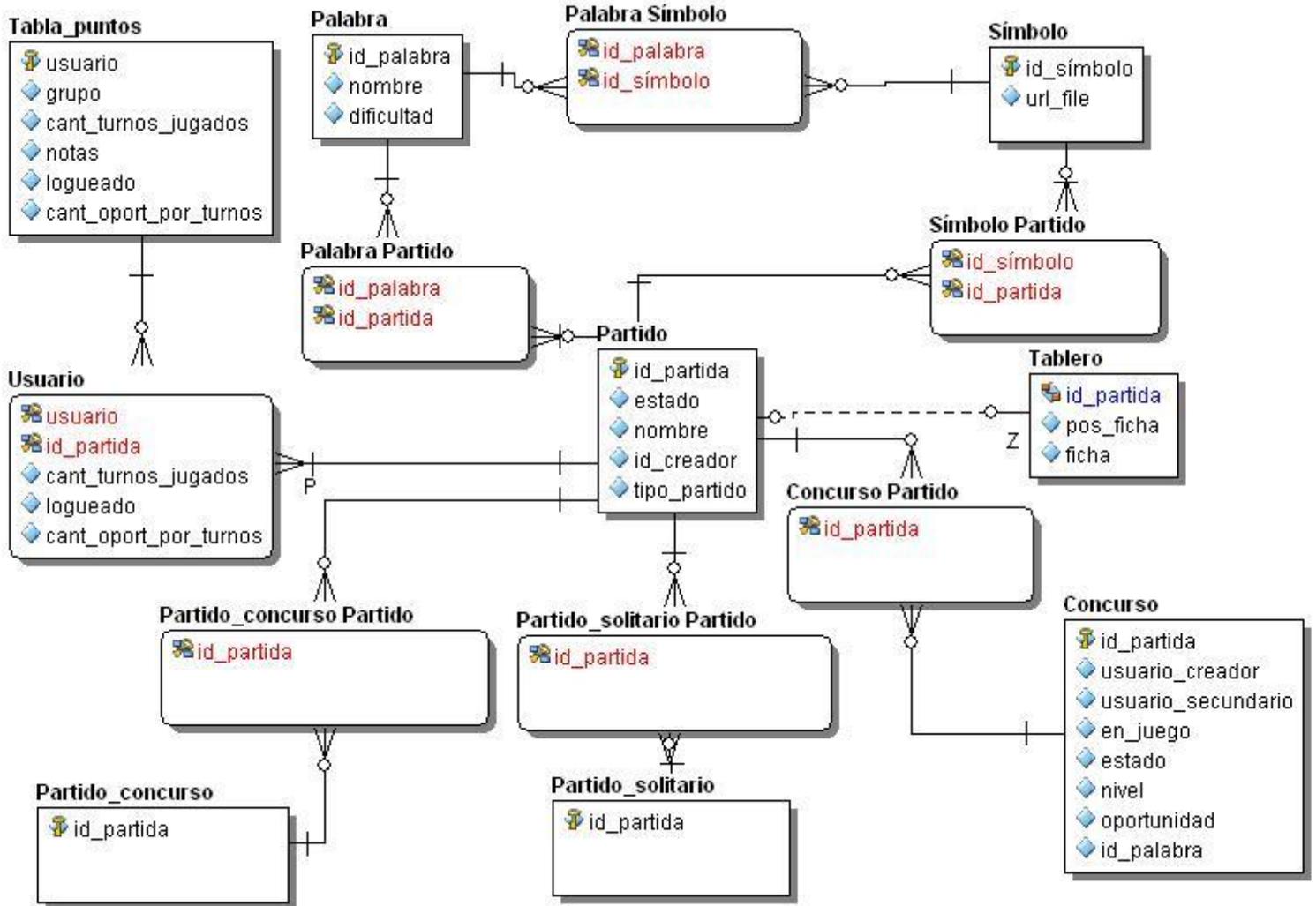


Figura 7 Modelo de datos.

### 1.6 Patrón de Arquitectura Modelo Vista Controlador (MVC).

MVC es un patrón de diseño de arquitectura utilizado en sistemas Web para separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos, permitiendo flexibilidad y facilidad a la hora de hacer futuros cambios.

El **Modelo** representa las estructuras de datos. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.

La **Vista** es la encargada de presentar la información al usuario. Una vista puede ser una página Web o parte de una página.

El **Controlador** actúa como intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para generar una página, es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo.

Teniendo en cuenta las características de la aplicación que se quiere desarrollar, que debe incluir un juego online de dos usuarios se decidió aplicar este patrón, ya que los datos que serán comunes y permitirán la interacción entre ellos serán almacenados en la base de datos, la cual se accedería a través de la clase **Modelo**. De esta manera se solucionaría el problema de que puedan jugar y observar los cambios de su oponente casi en tiempo real. La **Vista** permitiría implementar la interfaz necesaria para este juego, de la que se espera sea agradable, sencilla y lo más dinámica posible. El **Controlador** sería el intermediario entre la **Vista** y el **Modelo**, además de que contendría las funciones principales y controlará la seguridad interna del sitio.

### 3.5 Conclusiones

En el presente capítulo con el análisis y modelado del sistema, se obtuvieron los datos y reglas que en él serán manejados para su correcto funcionamiento, se establece una estructura interna y se alcanzó la primera aproximación de la implementación.

## CAPÍTULO 4: Implementación.

### 4.1 Introducción.

En este capítulo se describirá cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue, con el uso de las clases activas y subsistemas encontrados durante el diseño.

### 4.2 Modelo de implementación.

#### 4.2.1 Diagrama de despliegue.

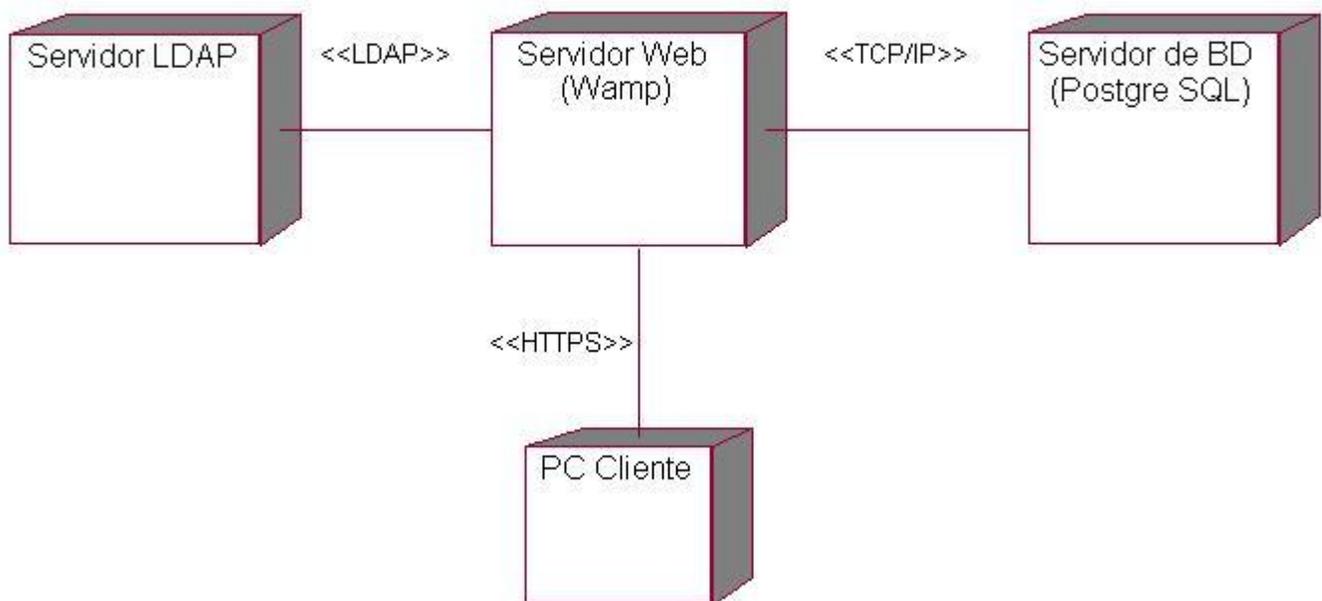


Figura 8 Diagrama de despliegue.

### 4.2.2 Diagramas de componentes.

#### Caso de uso “Ver notas”.

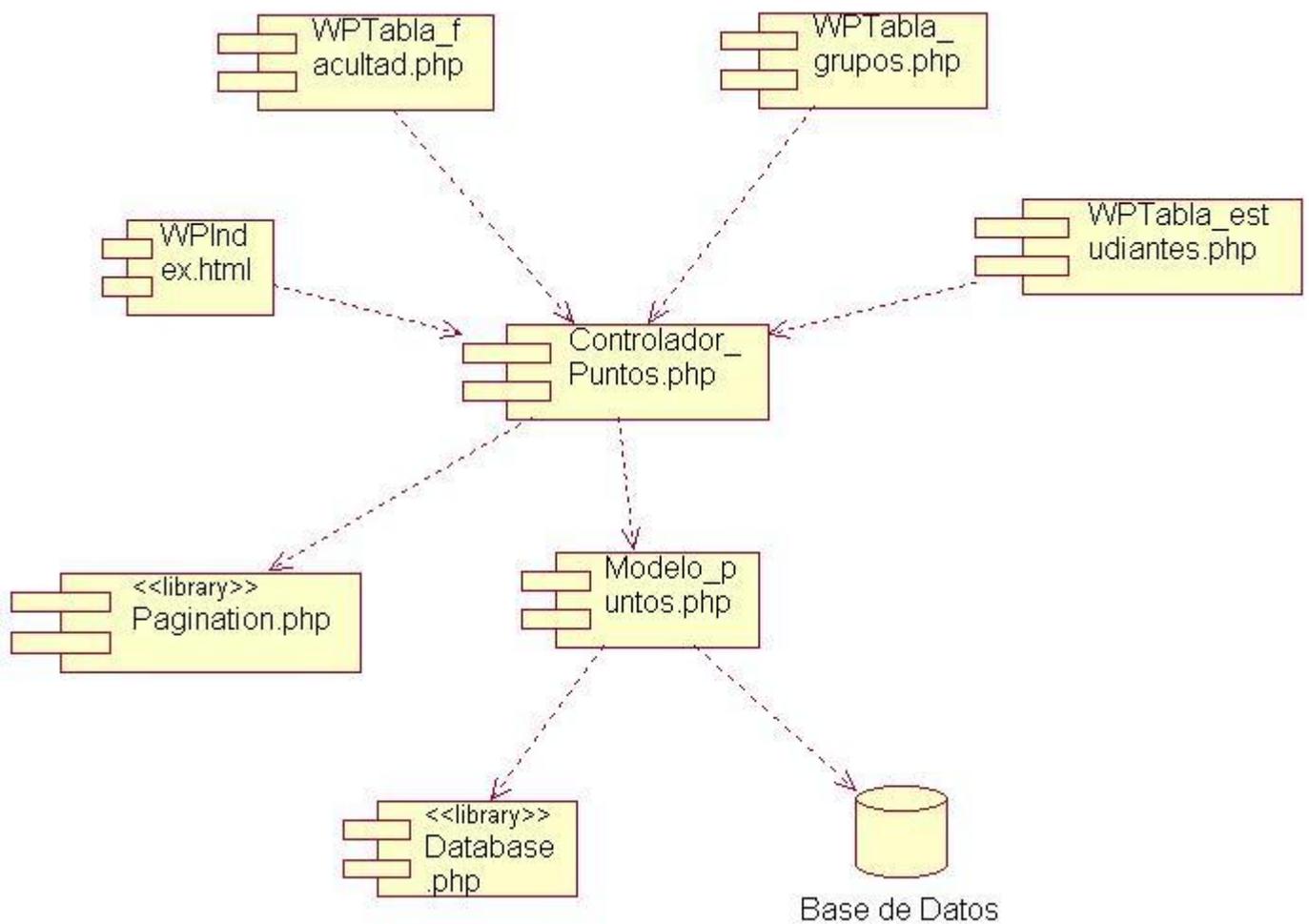
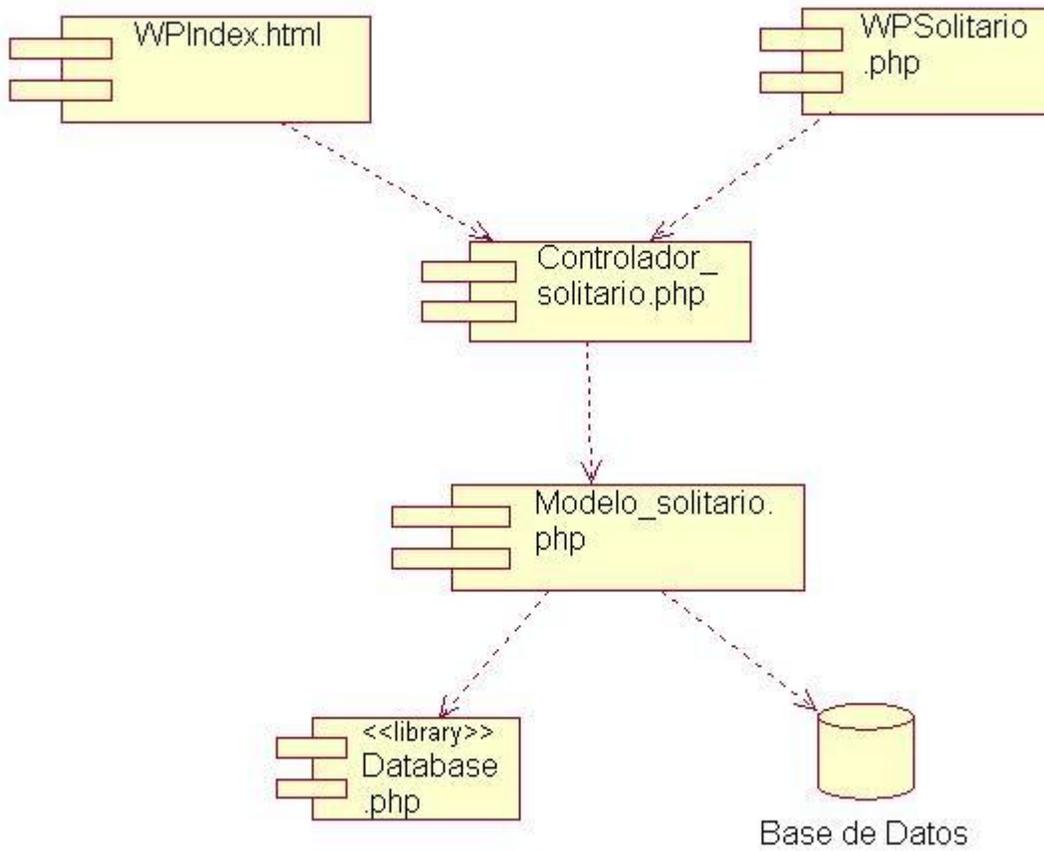


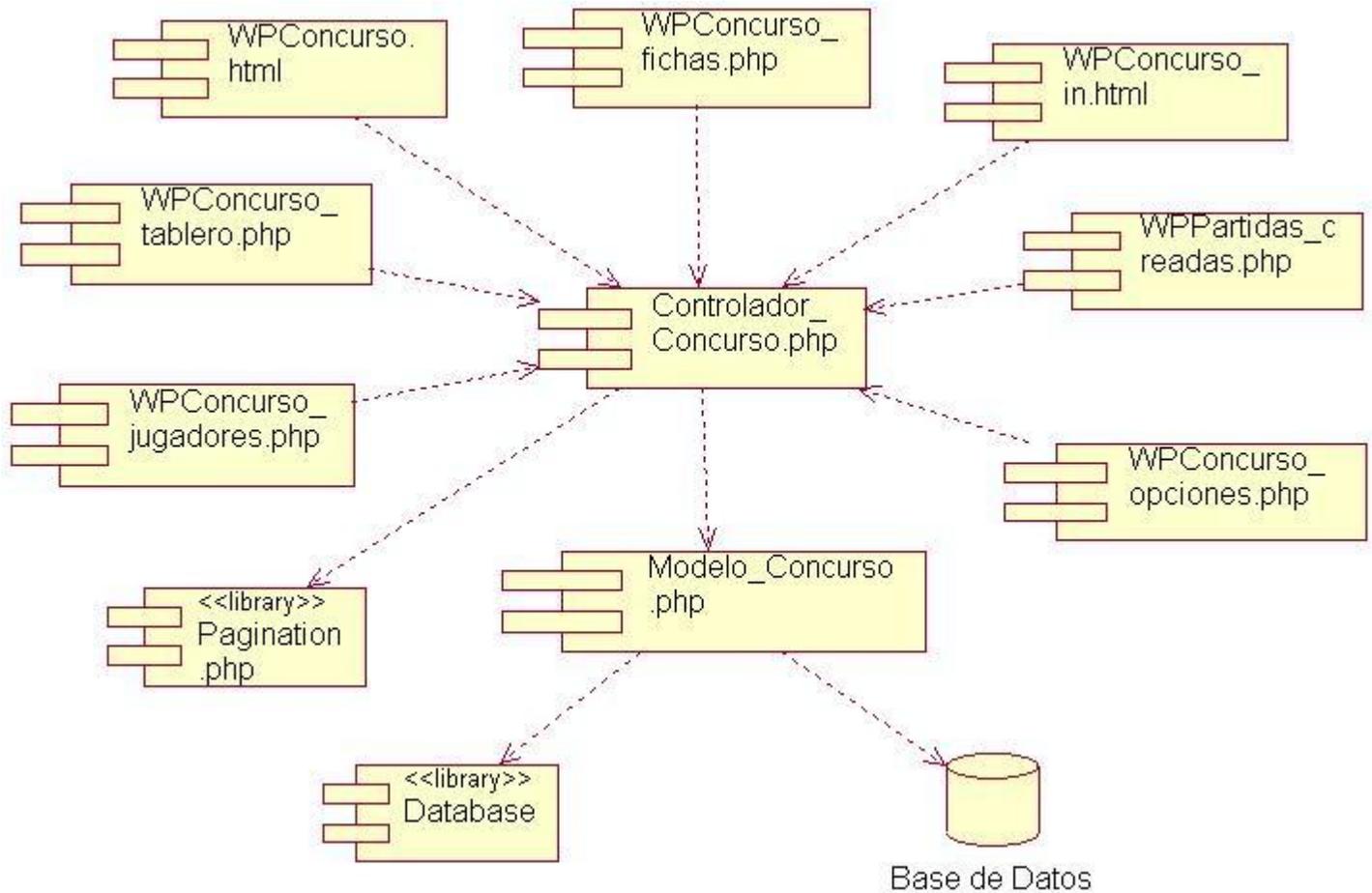
Figura 9 Diagrama de componentes CU “Ver notas”.

**Caso de uso “Solitario”.**



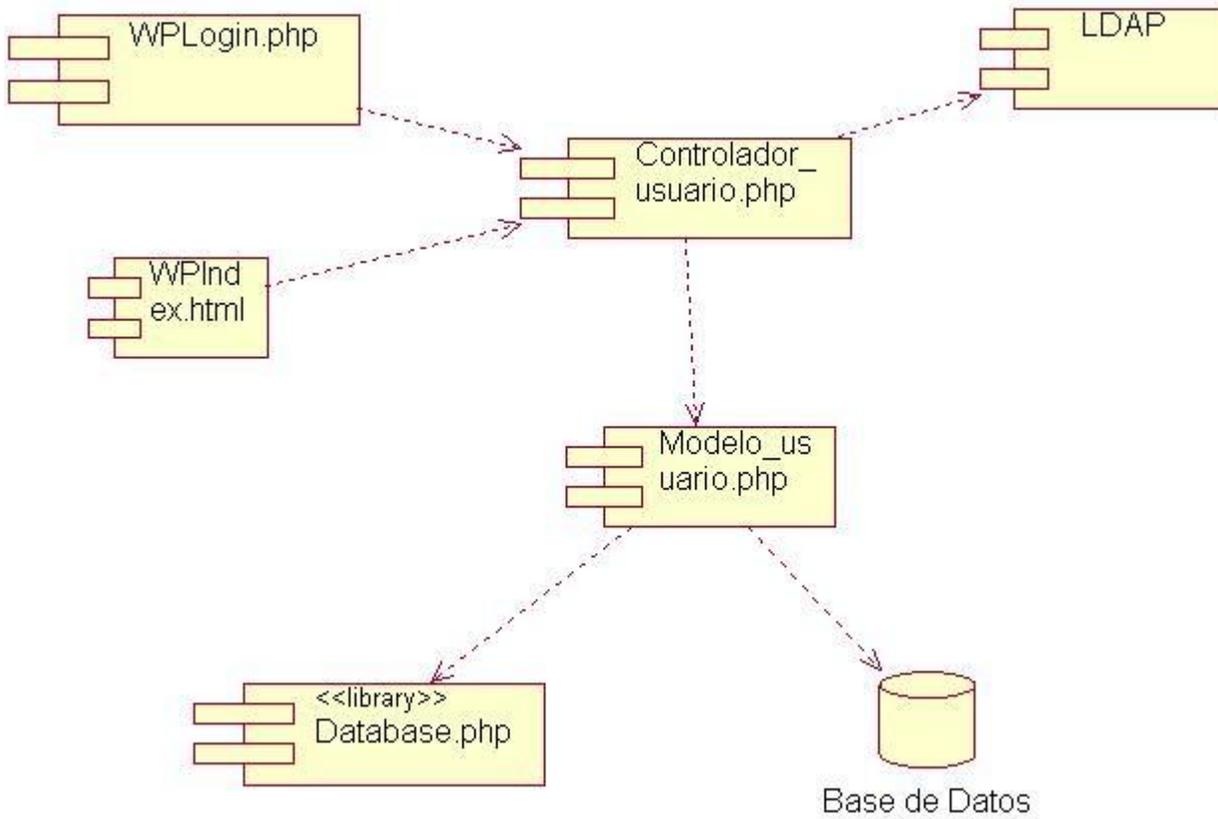
**Figura 10 Diagrama de componentes CU “Solitario”.**

**Caso de uso “Concurso”.**



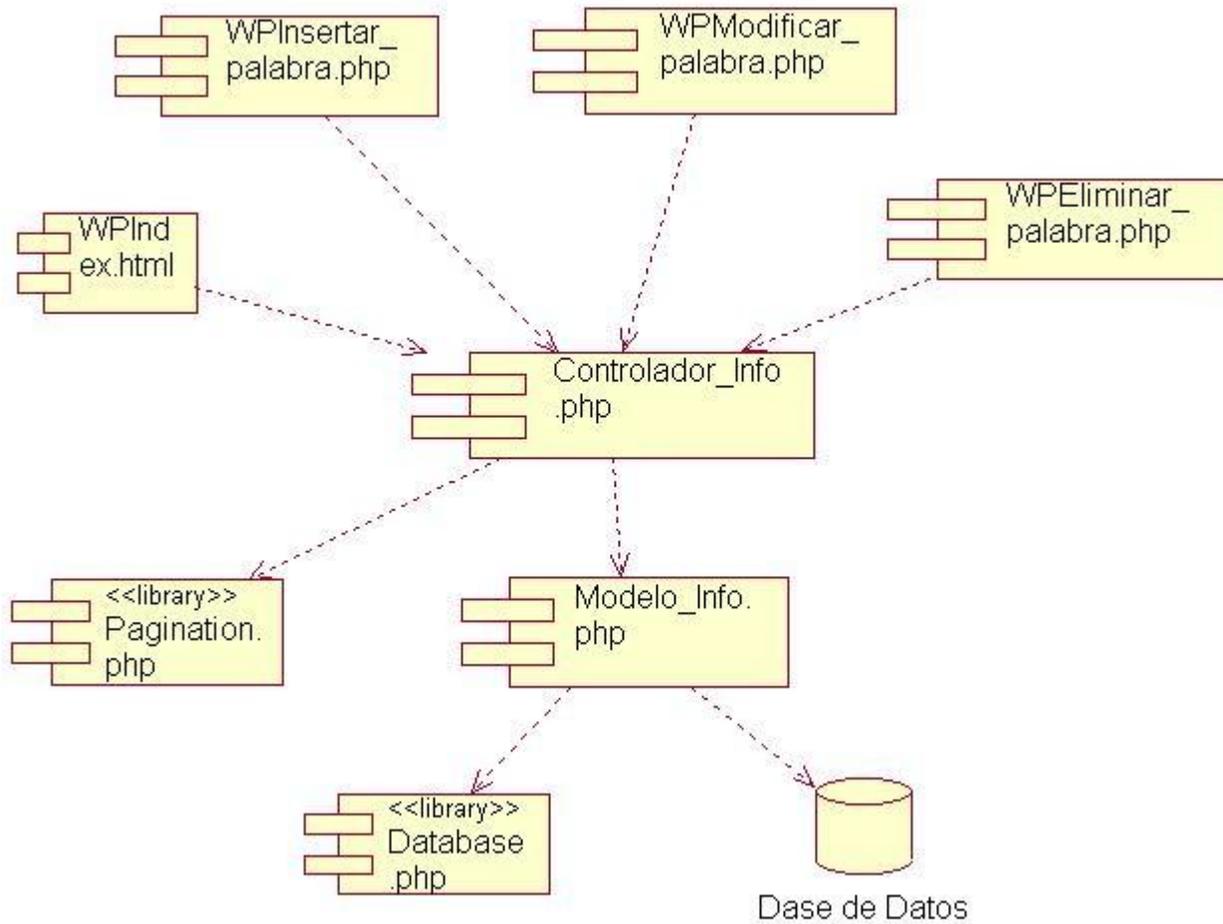
**Figura 11 Diagrama de componentes CU “Concurso”.**

**Caso de uso “Autenticar”.**



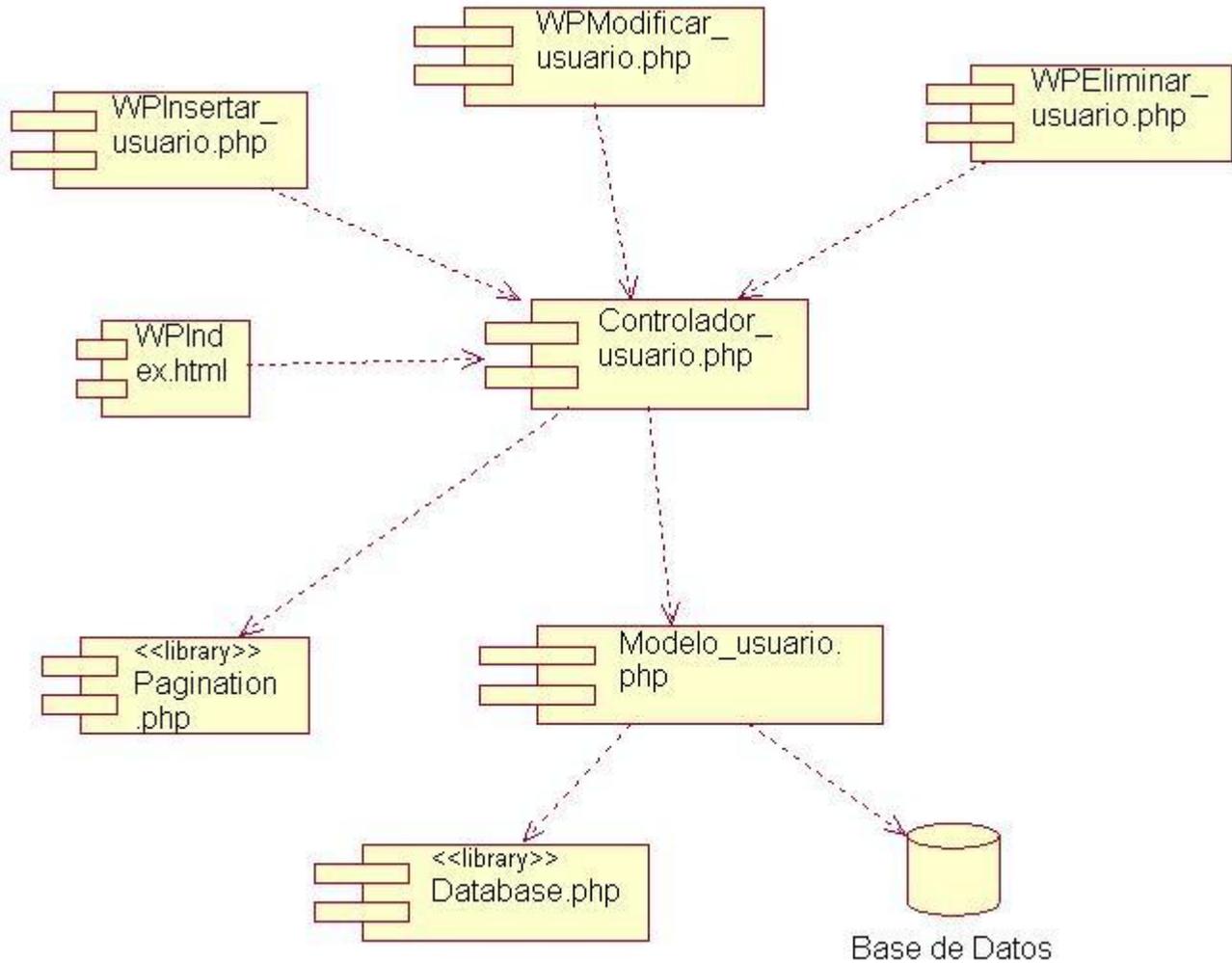
**Figura 12 Diagrama de componentes CU “Autenticar”.**

**Caso de uso “Gestionar Información”.**



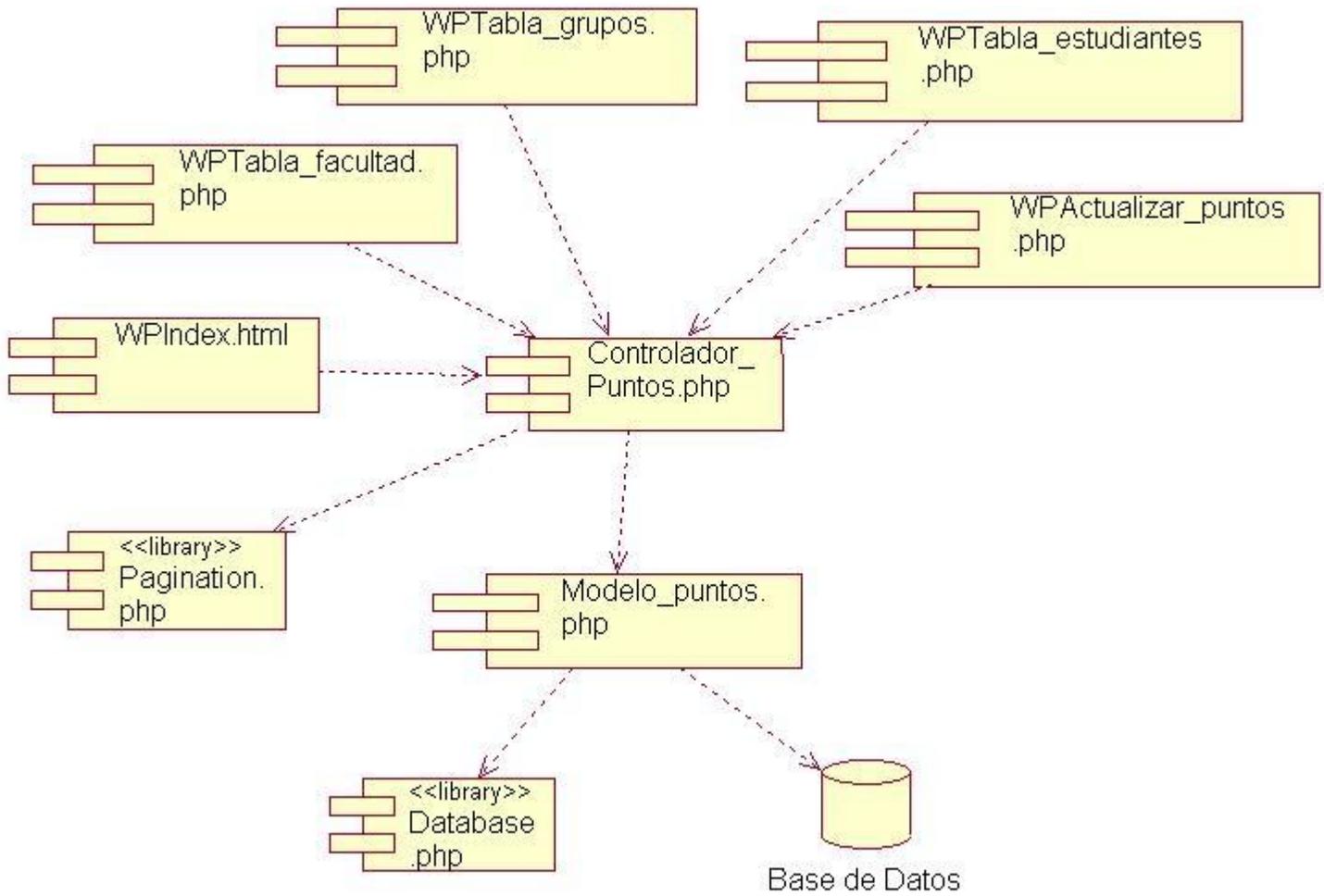
**Figura 13 Diagrama de componentes CU “Gestionar información”.**

**Caso de uso “Gestionar Usuarios”.**



**Figura 14 Diagrama de componentes CU “Gestionar usuarios”.**

**Caso de uso “Gestionar Puntos”.**



**Figura 15 Diagrama de componentes CU “Gestionar Puntos”.**

## **4.6 Conclusiones**

En el presente capítulo se abordaron las etapas de implementación y pruebas del software en desarrollo. Para ello se exponen todos los artefactos generados, realizando una descripción de cada uno de ellos.

## **CAPÍTULO 5: Estudio de Factibilidad.**

### **5.1 Introducción.**

En este capítulo, se realiza el cálculo de la estimación del trabajo mediante el Método de Estimación por Puntos de Casos de Uso. Para así poder determinar si es factible o no su implementación.

### **5.1 Introducción.**

En este capítulo, se realiza el cálculo de la estimación del trabajo mediante el Método de Estimación por Puntos de Casos de Uso. Para así poder determinar si es factible o no su implementación.

### **6.1 Aplicar Método de Estimación Puntos por Casos de Uso.**

La estimación mediante el análisis de Puntos de Casos de Uso es un método de estimación del tiempo de desarrollo de un proyecto, mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

### **6.2 Cálculo de Puntos de Casos de Uso sin ajustar.**

Los puntos de casos de uso sin ajustar se calculan a partir de la siguiente ecuación:

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de casos de uso sin ajustar

UAW: Factor de peso de los actores sin ajustar

UUCW: Factor de peso de los casos de uso sin ajustar

Factor de Peso de los Actores sin ajustar (UAW):

El Factor de Peso de los Actores sin ajustar se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los mismos se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

Tipo de Actor	Descripción	Factor de Peso	Cantidad de Actores	Cant*Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación.(API: Application Programming Interface)	1	0	0*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto	2	0	0*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica	3	3	3*3
<b>Total</b>				<b>9</b>

Tabla 2 Factor de peso de los actores sin ajustar.

Actores del Sistema	Tipo de Actor	Factor de Peso
Administrador del Sistema	Complejo	3
Profesor	Complejo	3
Estudiante	Complejo	3

Tabla 3 Actores del Sistema para la estimación.

$$UAW = \sum \text{Cantidad de Actores} * \text{Factor de Peso}$$

$$UAW = 9$$

**Factor de Peso de los Casos de Uso sin ajustar (UUCW)**

El Factor de Peso de los Casos de Uso sin ajustar se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo.

Tipo de Caso de Uso	Descripción	Factor de Peso	Cantidad de Casos de Uso	Cant*Peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5	2	2*5
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10	3	3*10
Complejo	El Caso de Uso contiene más de 8 transacciones	15	2	2*15
<b>Total</b>				<b>70</b>

**Tabla 4 Factor de peso de los casos de uso sin ajustar.**

Casos de Uso	Tipo de Caso de Uso	Factor de Peso
Gestionar roles de usuarios.	Medio	10
Gestionar Puntos	Complejo	15
Gestionar Información de Juegos	Complejo	15
Consultar Información de Juegos	Simple	5
Consultar Puntos	Simple	5
Solitario	Medio	10
Concurso	Medio	10

**Tabla 5 Casos de Uso del Sistema para la estimación.**

$$UUCW = \sum \text{Cantidad de CU} * \text{Factor de Peso}$$

$$UUCW = 2*5 + 3*10 + 2*15 = 10 + 30 + 30$$

$$UUCW = 70$$

*Finalmente el cálculo de los puntos de casos de uso sin ajustar queda:*

$$UUCP = UAW + UUCW$$

$$UUCP = 9 + 70$$

$$UUCP = 79$$

### 6.3 Cálculo de puntos de caso de uso ajustados.

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

$$UCP = UUCP * TCF * EF$$

Donde:

- UCP: *Puntos de casos de uso ajustados*
- UUCP: *Puntos de casos de uso sin ajustar*
- TCF: *Factor de complejidad técnica*
- EF: *Factor de ambiente*

#### Factor de Complejidad Técnica (TCF)

Factor de Complejidad Técnica se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

Factor	Descripción	Peso	Valor	Peso * Valori
T1	Sistema distribuido	2	2	2*2
T2	Objetivos de performance o tiempo de respuesta	1	4	1*4
T3	Eficiencia del usuario final	1	3	1*3
T4	Procesamiento interno complejo	1	4	1*4
T5	El código debe ser reutilizable	1	4	1*4
T6	Facilidad de instalación	0.5	5	0.5*5
T7	Facilidad de uso	0.5	5	0.5*5

T8	Portabilidad	2	5	2*5
T9	Facilidad de cambio	1	4	1*4
T10	Concurrencia	1	2	1*2
T11	Incluye objetivos especiales de seguridad	1	4	1*4
T12	Provee acceso directo a terceras partes	1	5	1*5
T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	3	1*3
<b>Total</b>				<b>52</b>

**Tabla 6 Factor de complejidad técnica**

$$TCF = 0.6 + 0.01 * \sum (\text{Peso}_i * \text{Valor}_i)$$

$$TCF = 0.6 + 0.01 * (2*2 + 1*4 + 1*3 + 1*4 + 1*4 + 0.5*5 + 0.5*5 + 2*5 + 1*4 + 1*2 + 1*4 + 1*5 + 1*3)$$

$$TCF = 0.6 + 0.01 * 52 = 0.6 + 0.52$$

$$TCF = 1.12$$

Factor de ambiente (EF)

El factor de ambiente está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor de 0 a 5 al igual que en el Factor de Complejidad Técnica.

Factor	Descripción	Peso	Valor	Peso <sub>i</sub> * Valor <sub>i</sub>
E1	Familiaridad con el modelo de proyecto utilizado	1.5	2	1.5*2
E2	Experiencia en la aplicación	0.5	4	0.5*4
E3	Experiencia en orientación a objetos	1	3	1*3
E4	Capacidad del analista líder	0.5	2	0.5*2
E5	Motivación	1	5	1*5

E6	Estabilidad de los requerimientos	2	3	2*3
E7	Personal part-time	-1	4	-1*4
E8	Dificultad del lenguaje de programación	-1	3	-1*3
<b>Total</b>				<b>13</b>

**Tabla 7 Factor de ambiente.**

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i * \text{Valor}_i)$$

$$EF = 1.4 - 0.03 * (1.5 * 2 + 0.5 * 4 + 1 * 3 + 0.5 * 2 + 1 * 5 + 2 * 3 + -1 * 4 + -1 * 3)$$

$$EF = 1.4 - 0.03 * 13 = 1.4 - 0.39$$

$$EF = 1.01$$

*Finalmente el cálculo de los puntos de casos de uso ajustados queda:*

$$UCP = UUCP * TCF * EF$$

$$UCP = 79 * 1.12 * 1.01$$

$$UCP = 89.3648$$

#### **6.4 Cálculo del Esfuerzo.**

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CF$$

Donde:

- E: *Esfuerzo estimado en horas-hombre*
- UCP: *Puntos de Casos de Uso ajustados*
- CF: *Factor de conversión*

Factor de conversión (CF)

Para obtener el factor de conversión se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8).

- Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de casos de uso.
- Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de casos de uso.
- Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

$$\text{Total}_{EF} = \text{Cant EF} < 3 \text{ (entre E1 –E6)} + \text{Cant EF} > 3 \text{ (entre E7, E8)} = 2 + 1 = 3$$

*Entonces si*  $\text{Total}_{EF} = 3$

$$CF = 28 \text{ Horas-Hombre} / \text{Punto de casos de uso}$$

*Finalmente el esfuerzo en Horas-Hombre queda:*

$$E = UCP * CF$$

$$E = 89.3648 * 28 \text{ Horas-Hombre}$$

$$E = 2502.21 \text{ Horas-Hombre}$$

### 6.5 Distribución del Esfuerzo entre las diferentes actividades de un proyecto:

Actividad	Porcentaje	Horas-Hombre
Análisis	20%	500.442 Horas-Hombre
Diseño	20%	500.442 Horas-Hombre
Implementación	40%	1000.884 horas-hombre
Prueba	10%	250.221 Horas-Hombre
Sobrecarga	10%	250.221 Horas-Hombre

<b>Total</b>	<b>100%</b>	<b>2502.21 Horas-Hombre</b>
--------------	-------------	-----------------------------

Tabla 8 Esfuerzo del proyecto

Suponiendo que una persona trabaje 8 horas por día, y un mes tiene como promedio 30 días; la cantidad de horas que puede trabajar una persona en 1 mes es 240 horas.

Si  $E = 2502.21$  horas - hombre y por cada 240 horas yo tengo un mes eso daría un  $E = 10.43$  mes - hombre.

Esto quiere decir que 1 persona puede realizar el software en más o menos 10 meses.

Si en el equipo hay 2 personas y todas realizan el mismo esfuerzo entonces el problema analizado puede terminarse en aproximadamente 5 meses.

## 7.2 Análisis de Costo.

Costo del Proyecto.

*Se asume como salario promedio mensual \$100.00*

CH: *Cantidad de hombres*

CHM: *Costo por hombres - mes*

Tiempo: *Tiempo total del proyecto*

E: *Esfuerzo*

$CH = 2$  *hombres*

$CHM = 2 * \text{Salario Promedio} = \$200.00$  *mes*

$\text{Costo} = CHM * E / CH$

$\text{Costo} = 200.00 * 10 / 2$

$\text{Costo} = \$1000$

$$\text{Tiempo} = E / CH$$

$$\text{Tiempo} = 10 / 2$$

$$\text{Tiempo} = 5$$

De los resultados obtenidos se interpreta que con 2 hombres trabajando en el software, el mismo es desarrollado en 5 meses aproximadamente y su costo total se estima que sea \$1000.

### **7.3 Conclusiones.**

En este capítulo se describió el estudio de factibilidad realizado correspondiente al sistema propuesto, teniendo en cuenta el costo estimado y los beneficios que reportará al ser implantado. El estudio realizado ha proporcionado valiosos argumentos, que permiten llegar a la conclusión de que la solución de software es factible para su puesta en funcionamiento.

## CONCLUSIONES

Luego de analizar la problemática existente en la Universidad de las Ciencias Informáticas en cuanto a la enseñanza-aprendizaje del idioma inglés y la falta de motivación por parte de los estudiantes por esta asignatura, se determinó desarrollar una propuesta de un juego sobre fonética inglesa como solución. En el presente informe se describieron las herramientas necesarias para implementar la aplicación así como un breve estudio sobre aplicaciones similares en el mundo y en la universidad. De forma general se cumple el objetivo propuesto pues se crea “Phoname” un juego para motivar y desarrollar las habilidades para la fonética inglesa de los estudiantes de la Universidad.

Obteniéndose además los siguientes resultados:

1. Quedó definida la metodología para la construcción del software a partir del estudio realizado.
2. Se cumplió con las expectativas del cliente.
3. Quedó demostrada la validez y eficacia de las herramientas y tecnologías seleccionadas.

Por lo anteriormente descrito queda reflejado el cumplimiento de los objetivos específicos.

## **RECOMENDACIONES**

Con el objetivo de mejorar la enseñanza-aprendizaje de la pronunciación del idioma inglés en la facultad se recomienda:

- ✓ Agregar funcionalidades que permitan jugar a más de dos estudiantes a la vez.
- ✓ Desarrollar un sistema similar que emplee sonido para mejorar aún más las habilidades en la pronunciación.
- ✓ Extender el alcance de la propuesta de manera que pueda ser utilizada en otras universidades del país.

**REFERENCIAS BIBLIOGRÁFICAS**

- 1 English Club [En línea] [www.EnglishClub.com](http://www.EnglishClub.com)
- 2 Cambridge English Online [En línea] [www.CambridgeEnglishOnline.com](http://www.CambridgeEnglishOnline.com)
- 3 Php ya [En línea] [www.phpya.com.ar](http://www.phpya.com.ar)
- 4 Tesis Módulo de Control de Hardware del SIGLAB Junio 2008
- 5 Html point [En línea] [www.htmlpoint.com](http://www.htmlpoint.com)
- 6 Rumbaugh, James, Jacobson, Ivar and Booch, Grady. 2000. El Lenguaje Unificado de Modelado. [En línea] 2000. [Cited: Enero 10, 2008.]  
<http://bibliodoc.uci.cu/pdf/reg00060.pdf>.
- 7 Aprender Inglés Gratis, Cambridge English Online, [En línea] Por [Luís Martínez González](#), en 25 de Abril de 2009  
<http://www.aprendergratis.com/cambridge-english-online.html>
- 8 Eguíluz Pérez, Javier. 2007. Libros Web. Introducción a AJAX. [En línea] 2007.  
<http://www.librosweb.es/ajax>.
- 9 ¿Qué es javascript? [En línea] [http://74.125.47.132/search?q=cache:-HlgGFkkwXAJ:geneura.ugr.es/~victor/cursillos/javascript/js\\_intro.html+javascript+es+orientado+a+eventos&cd=3&hl=es&ct=clnk&gl=cu](http://74.125.47.132/search?q=cache:-HlgGFkkwXAJ:geneura.ugr.es/~victor/cursillos/javascript/js_intro.html+javascript+es+orientado+a+eventos&cd=3&hl=es&ct=clnk&gl=cu)
- 22 Web oficial de CodeIgniter. [En línea] 04 30, 2008. <http://codeigniter.com>.

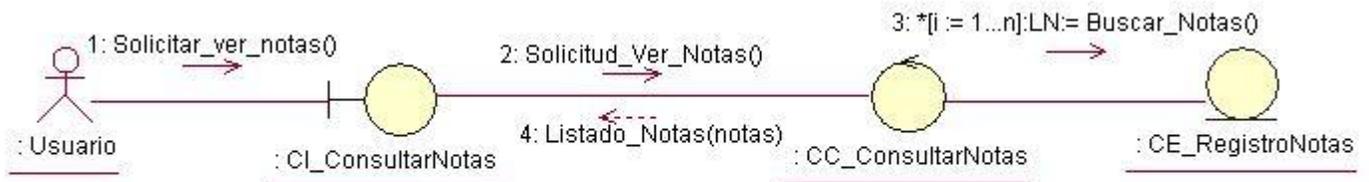
**BIBLIOGRAFÍA**

- 1 Tesis Módulo de Control de Hardware del SIGLAB Junio 2008.
- 2 *El lenguaje HTML*. [En línea] 2008-2009. Obtenido de <http://www.juntadeandalucia.es/averroes/iesgaviota/informatica/html.html>
- 6 Rumbaugh, James, Jacobson, Ivar and Booch, Grady. 2000. *El Lenguaje Unificado de Modelado*. [En línea] 2000. <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.
- 7 Aprender Inglés Gratis, Cambridge English Online, Por [Luís Martínez González](#), [En línea] 25 de Abril de 2009 Obtenido de <http://www.aprendergratis.com/cambridge-english-online.html>
- 8 Eguíluz Pérez, Javier. 2007. *Libros Web. Introducción a AJAX*. [En línea] 2007. Obtenido de <http://www.librosweb.es/ajax>.
- 9 ¿Qué es javascript? [En línea] Obtenido de [http://74.125.47.132/search?q=cache:-HlqGFkkwXAJ:geneura.ugr.es/~victor/cursillos/javascript/js\\_intro.html+javascript+es+orientado+a+eventos&cd=3&hl=es&ct=clnk&gl=cu](http://74.125.47.132/search?q=cache:-HlqGFkkwXAJ:geneura.ugr.es/~victor/cursillos/javascript/js_intro.html+javascript+es+orientado+a+eventos&cd=3&hl=es&ct=clnk&gl=cu)
- 10 *CodeIgniter User Guide Version 1.7.1*. [En línea] 2006-2008.
- 11 Jacobson, I., Booch, G., & Rumbaugh, J. [En línea] 1998. *El lenguaje unificado de modelado. Manual de referencia*.
- 12 Jacobson, I., Booch, G., & Rumbaugh, J. [En línea] 2000. *El proceso unificado de desarrollo del software*.
- 13 Optimiza CodeIgniter [En línea] Obtenido de <http://www.tufuncion.com/codeigniter-php>

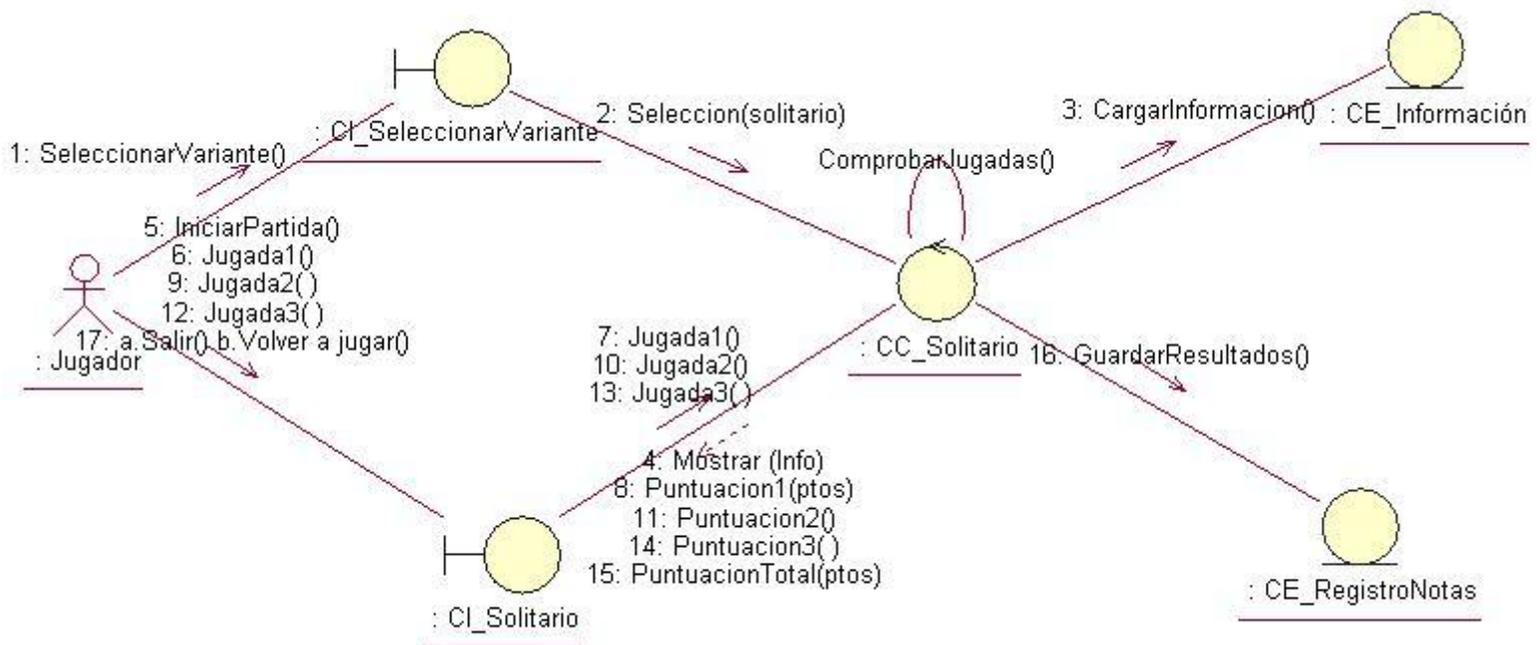
- 14 *Capítulo 1: Introducción a JavaScript*. [En línea] 2008. Obtenido de <http://www.elcodigo.net/tutoriales/javascript/javascript1.html>

**ANEXOS**

**Anexo 1: Diagramas de Colaboración.**



**Figura 16 Diagrama de colaboración CU "Ver Notas".**



**Figura 17 Diagrama de colaboración CU "Solitario".**

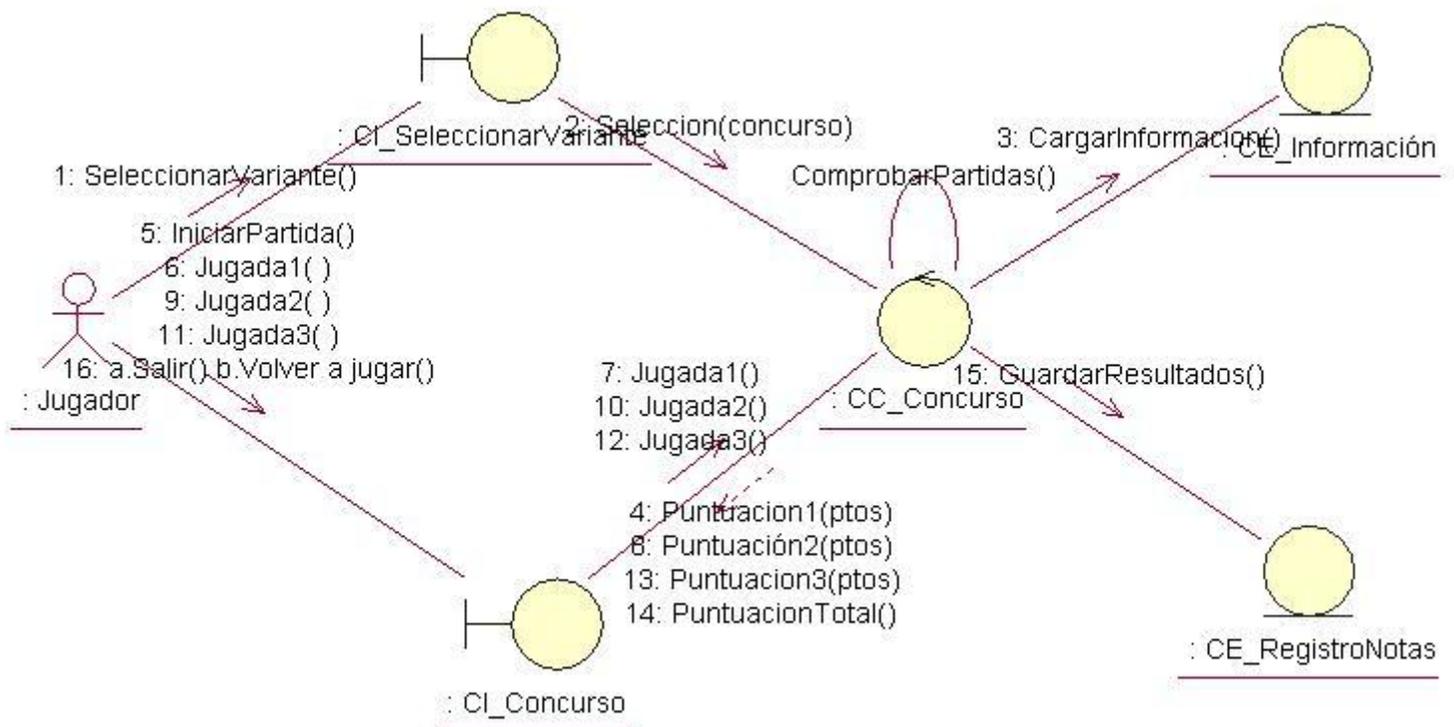


Figura 18 Diagrama de colaboración CU "Concurso".

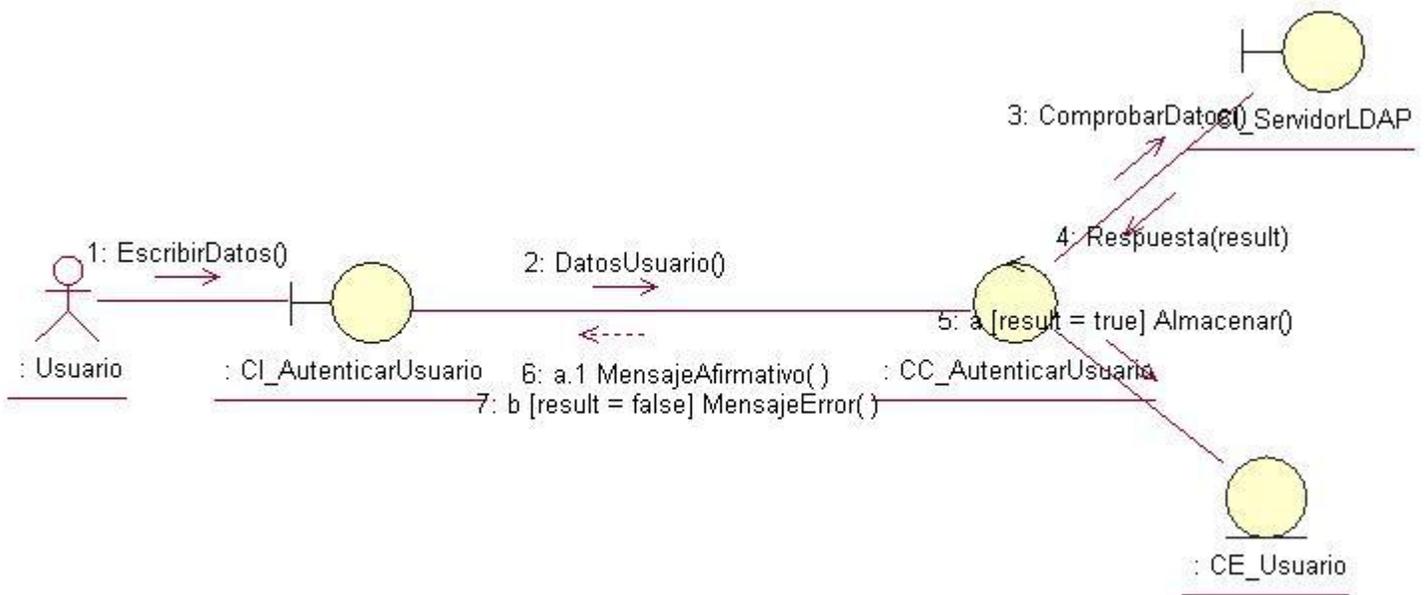
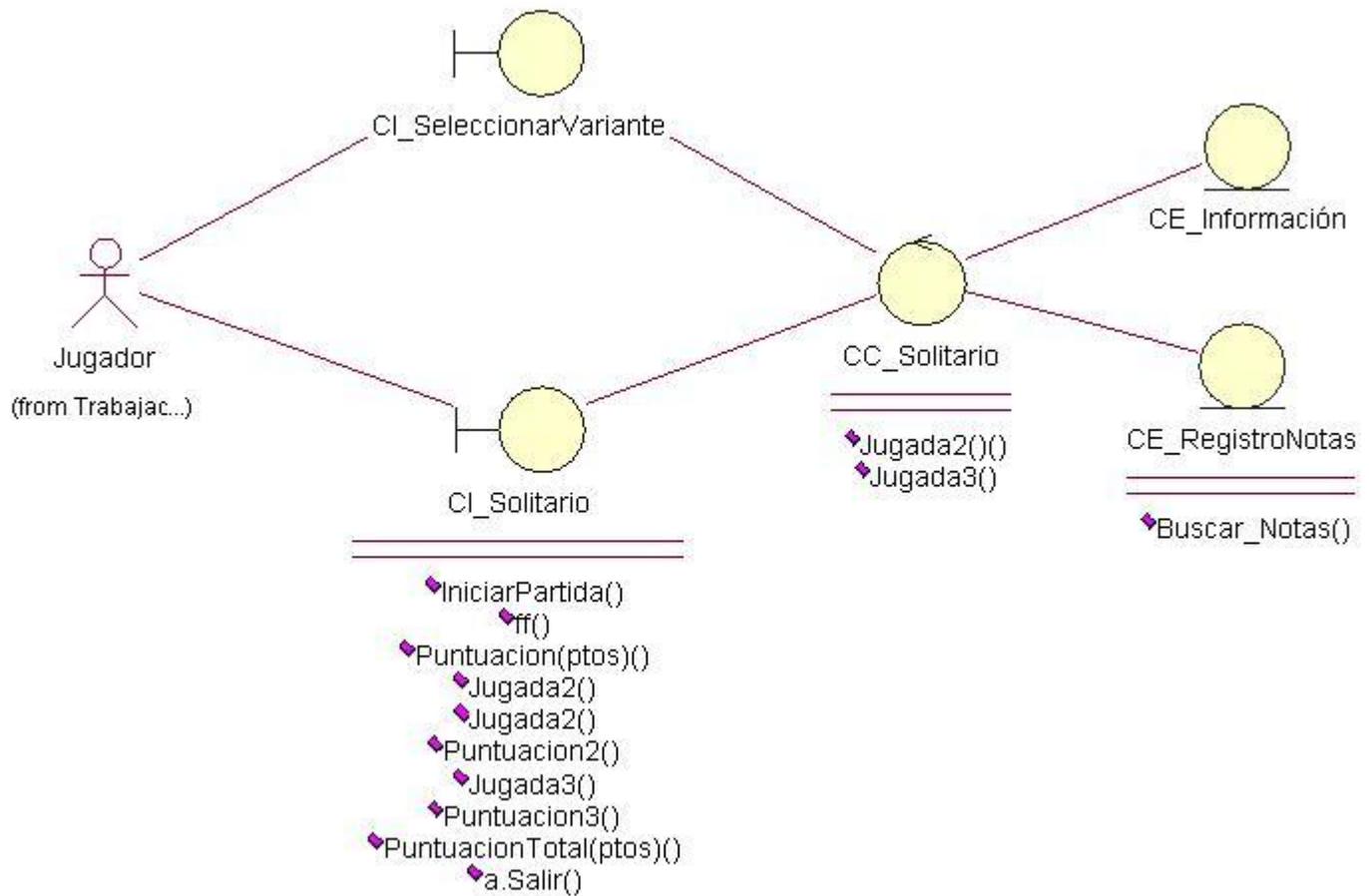


Figura 19 Diagrama de colaboración CU “Autenticar”.



Figura 20 Diagrama de clase del análisis CU “Consultar Puntos”.



**Figura 21 Diagrama de clase del análisis CU "Solitario".**

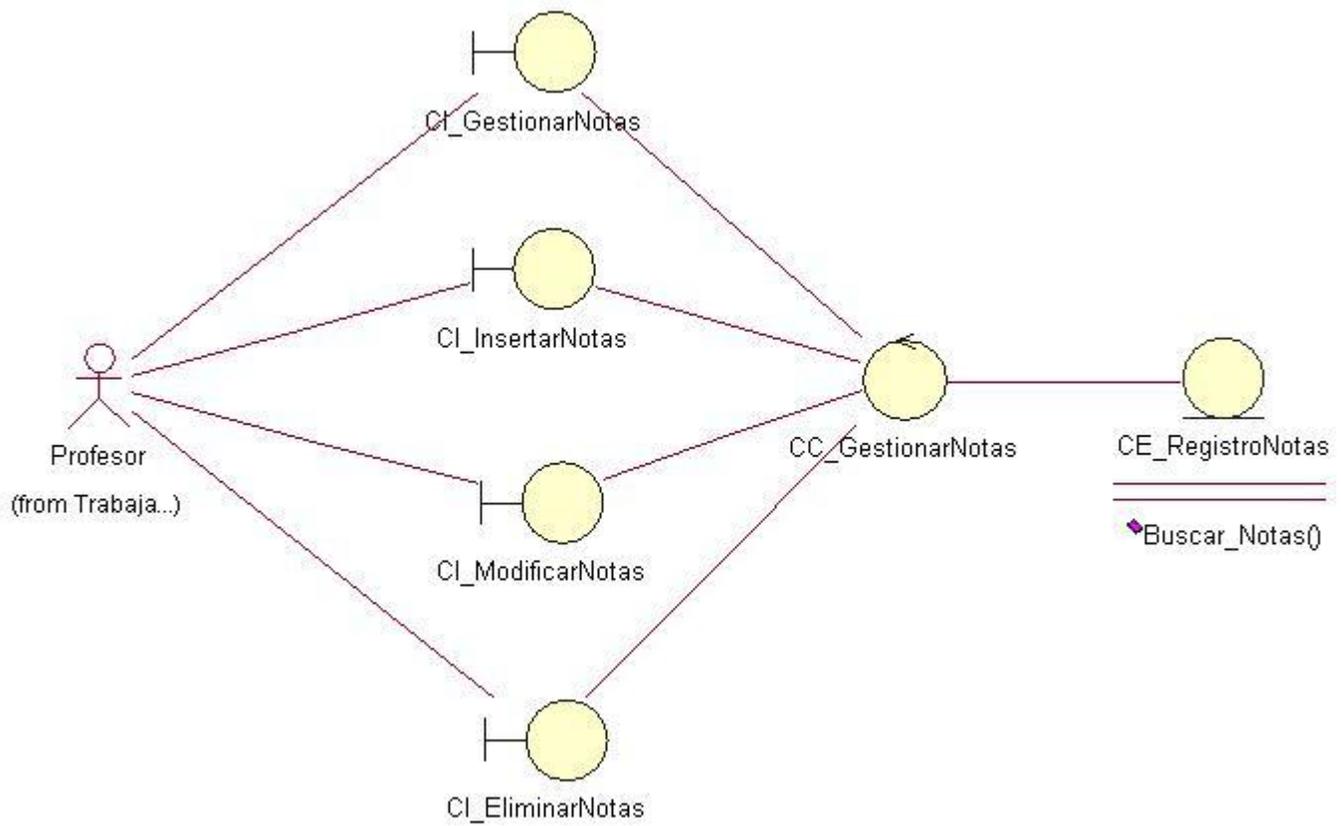
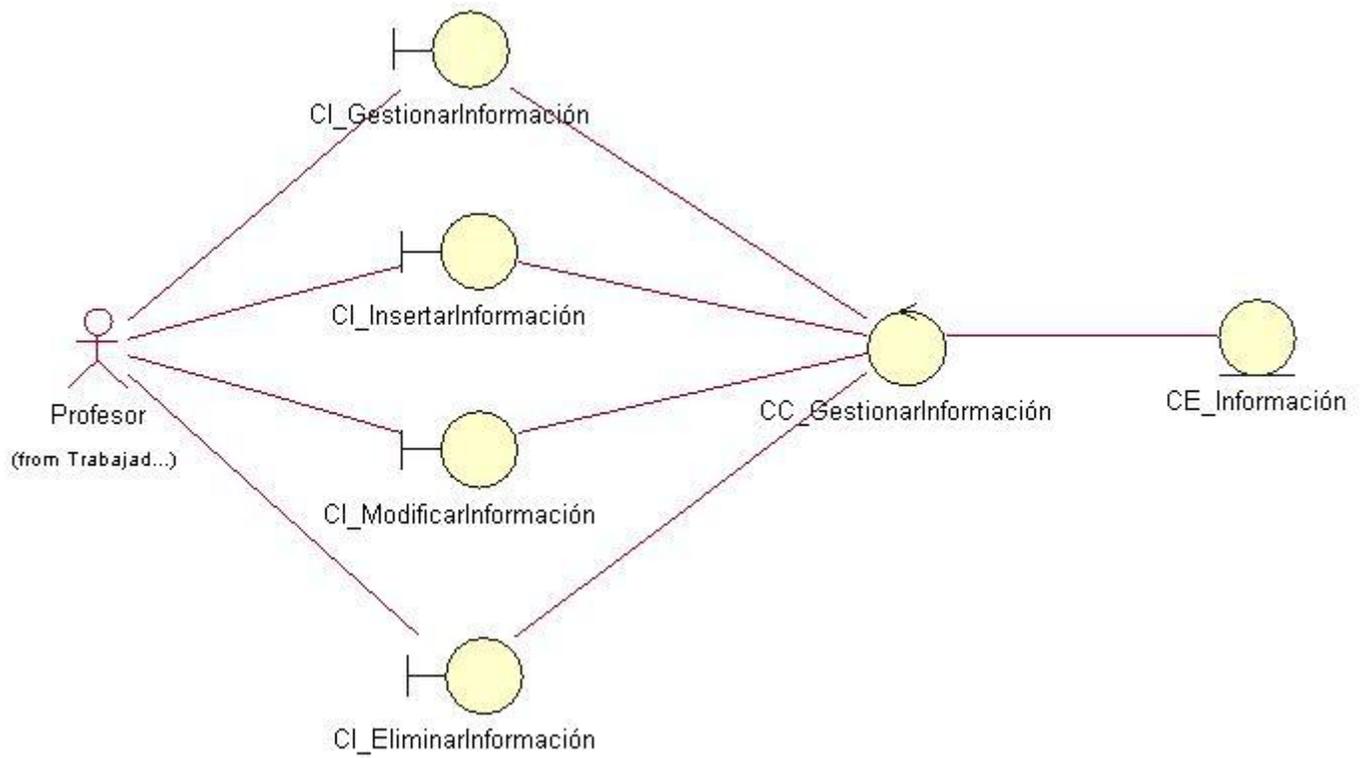


Figura 22 Diagrama de clase del análisis CU "Gestionar Puntos".



**Figura 23 Diagrama de clase del análisis CU "Gestionar información".**

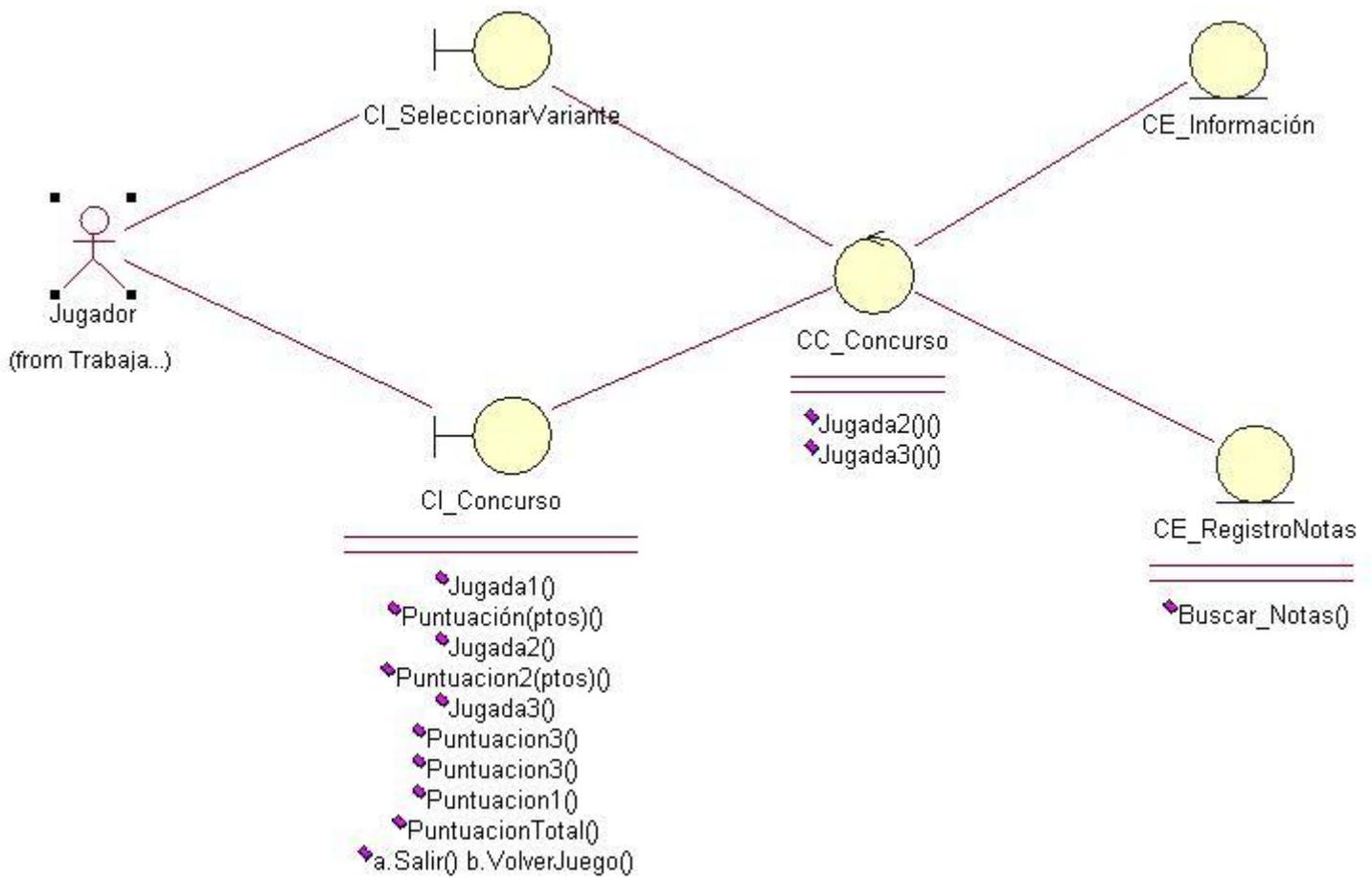


Figura 24 Diagrama de clase del análisis CU "Concurso".

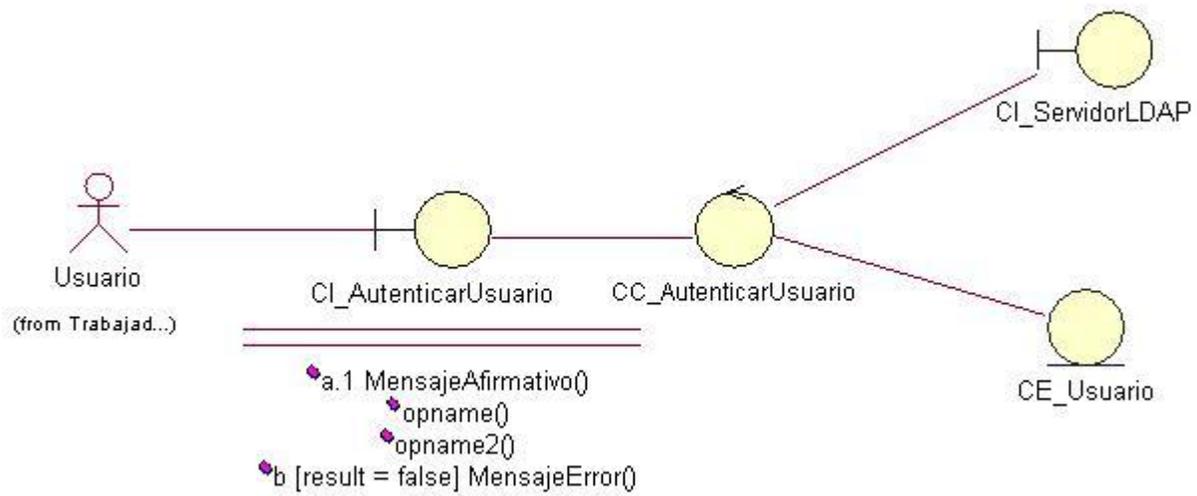


Figura 25 Diagrama de clase del análisis CU "Autenticar".

**Anexo 2: Descripción textual de los casos de uso del sistema.**

<b>Caso de Uso:</b>	Solitario	
<b>Actores:</b>	Jugador	
<b>Resumen:</b>	El caso de uso se inicia cuando el jugador selecciona la opción jugar solitario.	
<b>Precondiciones:</b>	Debe tener una cuenta en el dominio UCI.	
<b>Referencias:</b>		
<b>Prioridad:</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El jugador selecciona empezar partida.	1.1 El sistema le muestra la interfaz del juego. 1.2 El sistema muestra la palabra a transcribir.	
1.3 El jugador transcribe la palabra y elige comprobar.	1.4 El sistema le muestra un mensaje de que la transcripción es correcta y acumula los puntos.	
1.5 El jugador tiene dos oportunidades más.		

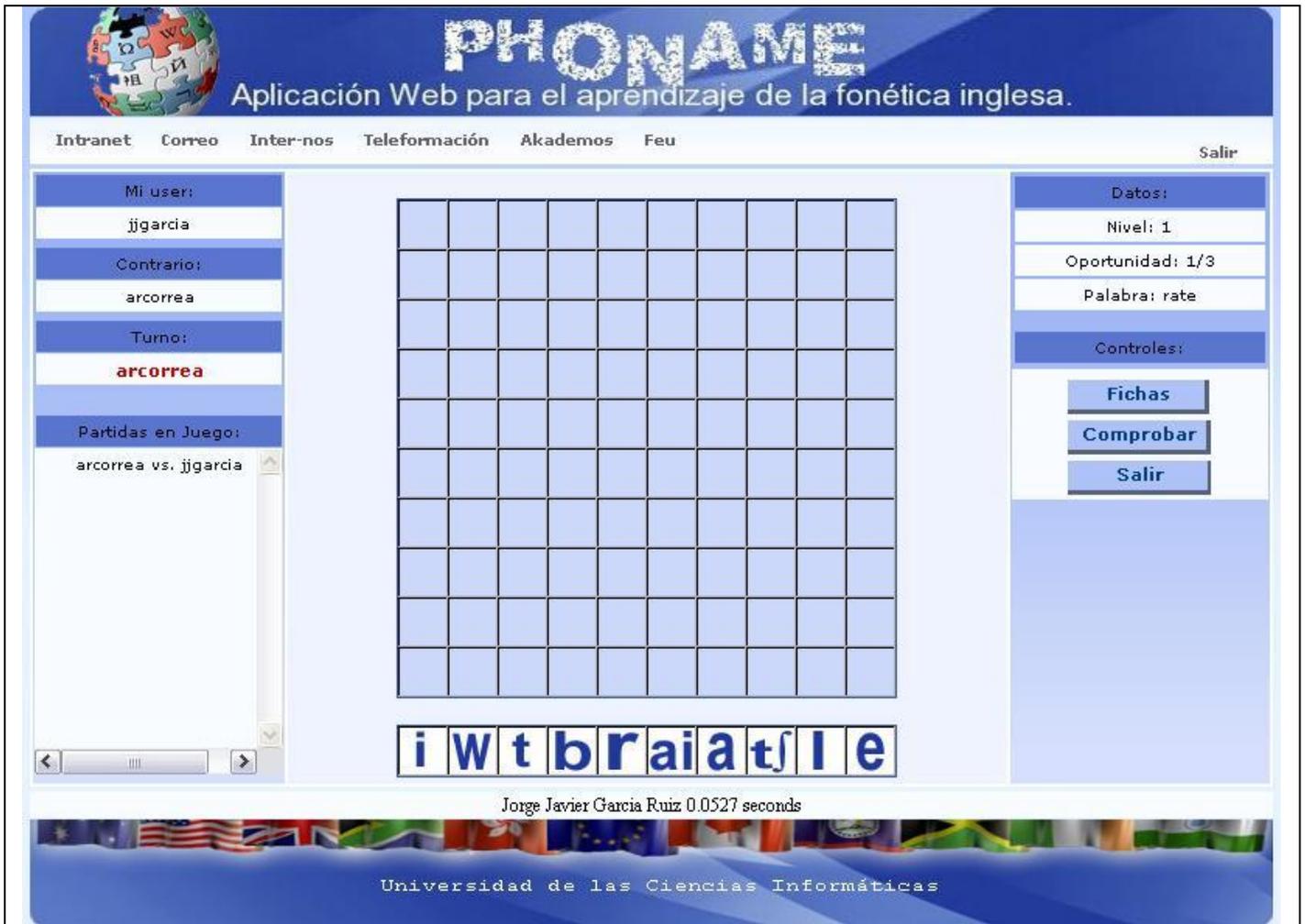
**Flujos Alternativos de Eventos**

Acción del Actor	Respuesta del Sistema
	1.4 Si la transcripción es incorrecta muestra un mensaje al usuario de que se equivocó.  1.5 El sistema no acumula puntos.

<b>Poscondiciones</b>	Debe quedar almacenado en la base de datos los puntos del usuario.
-----------------------	--

Tabla 9 Descripción textual caso de uso de sistema “Solitario”.

<b>Caso de Uso:</b>	Concurso	
<b>Actores:</b>	Jugador	
<b>Resumen:</b>	El caso de uso se inicia cuando el jugador elige la opción concurso y crea una partida o se une a una.	
<b>Precondiciones:</b>	Debe tener una cuenta en el dominio UCI.	
<b>Referencias:</b>		
<b>Prioridad:</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El jugador crea una partida.	1.1 El sistema le muestra una interfaz de espera hasta que se una otro jugador. 1.2 El sistema muestra la interfaz correspondiente al juego concurso. 1.3 El sistema le muestra al creador de la partida una palabra.	
1.5 El creador de la partida intenta transcribir la palabra y elige comprobar.	1.6 El sistema le muestra un mensaje de confirmación de que la palabra está correcta.	
1.7 Es el turno del jugador que se unió a la partida. 1.8 Cada jugador tiene tres oportunidades.		



**Flujos Alternativos de Eventos**

Acción del Actor	Respuesta del Sistema
	1.6 Si la transcripción es incorrecta muestra un mensaje al usuario de que se equivocó.
<b>Poscondiciones</b>	1.7 El sistema no acumula puntos.

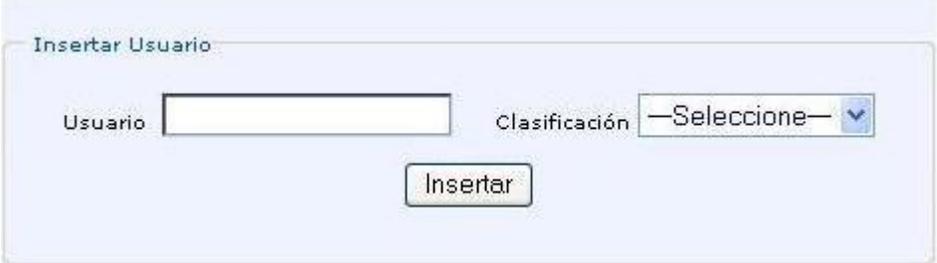
Tabla 10 Descripción textual caso de uso “Concurso”.

<b>Caso de Uso:</b>	Gestionar Información de Juego.	
<b>Actores:</b>	Administrador	
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador selecciona la opción gestionar información de juego.	
<b>Precondiciones:</b>	Debe existir información en la base de datos.	
<b>Referencias:</b>		
<b>Prioridad:</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador accede a la opción del menú "Gestionar Información".	1.1 El sistema muestra la interfaz correspondiente a la gestión de información.	
1.2. El estudiante selecciona la acción a realizar. En caso de: <ul style="list-style-type: none"> <li>1. "Agregar nueva palabra" ir a sección 1.</li> <li>2. "Eliminar documento de tesis" ir a sección 2.</li> </ul>		
<b>Sección "Agregar nueva palabra"</b>		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador escribe la palabra. 1.1 El administrador selecciona el nivel de dificultad. 1.2 El administrador guarda su transcripción.	1.3 Muestra mensaje de confirmación de que la palabra ha sido almacenada.	
<b>Flujos Alternativos de eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	

3. El administrador presiona el botón "Cancelar".	3.1 El sistema muestra al usuario el menú principal.
<b>Sección "Eliminar información"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona la información a eliminar. 1.1 Presiona el botón eliminar.	1.2 Muestra la tabla actualizada.

**Tabla 11 Descripción textual caso de uso "Gestionar Información".**

<b>Caso de Uso:</b>	Gestionar roles de usuarios.
<b>Actores:</b>	Administrador de sistema.
<b>Resumen:</b>	El caso de uso se inicia cuando el Administrador accede al sistema para definir los diferentes tipos de roles y asignarle a cada uno los permisos necesarios. De esta forma se obtienen roles con sus respectivos permisos, los cuales controlarán el acceso de los usuarios a la información así como a las acciones a realizar en el sistema.
<b>Precondiciones:</b>	
<b>Referencias:</b>	
<b>Prioridad:</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador accede a la opción del menú "Gestionar Usuarios".	1.1 Muestra la interfaz correspondiente, enviando a la misma el listado de todos los roles creados hasta el momento y brinda todas las posibles acciones a ejecutar.

<p>2. El administrador selecciona la acción a realizar. En caso de:</p> <ol style="list-style-type: none"> <li>1. “Crear rol” ir a la sección 1.</li> <li>2. “Eliminar rol” ir a sección 2.</li> <li>3. “Modificar rol” ir a sección 3.</li> </ol>	
<b>Sección 1 “Insertar usuario”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El administrador oprime el botón inserta el usuario nuevo, su rol y presiona el botón insertar.</p>	<p>1.1 El sistema muestra la misma interfaz actualizada.</p>
	
<b>Flujos Alternativos de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<p>2.1 En caso de haber datos incorrectos o incompletos se muestra un mensaje de error indicando los campos incorrectos y se retorna al paso 2 finalizando el caso de uso.</p>
<b>Sección 2 “Eliminar Rol”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El administrador selecciona un rol y presiona el botón “eliminar”.</p>	<p>1.1 El sistema muestra la misma interfaz con los datos actualizados.</p>

<p>Actualizar-Eliminar Usuarios</p> <p><b>Listado de Usuarios</b></p> <table border="1"> <thead> <tr> <th>Seleccione</th> <th>Usuario</th> <th>Clasificación</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td>arcorrea</td> <td>Administrador</td> <td>Elim   Act</td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td>jjgarcia</td> <td>Administrador</td> <td>Elim   Act</td> </tr> </tbody> </table> <p style="text-align: center;">Eliminar Marcados</p>		Seleccione	Usuario	Clasificación	Acciones	<input type="checkbox"/>	arcorrea	Administrador	Elim   Act	<input type="checkbox"/>	jjgarcia	Administrador	Elim   Act
Seleccione	Usuario	Clasificación	Acciones										
<input type="checkbox"/>	arcorrea	Administrador	Elim   Act										
<input type="checkbox"/>	jjgarcia	Administrador	Elim   Act										
<b>Sección 3 “Modificar rol”</b>													
<b>Flujo Normal de Eventos</b>													
Acción del Actor	Respuesta del Sistema												
1. El administrador presiona el botón “Actualizar”.	1.1 El sistema muestra una interfaz con el campo de roles para que el administrador seleccione la que desea.												
1.2 El administrador selecciona el rol presiona actualizar.	1.3 El sistema muestra una interfaz con los datos actualizados.												
<p>Actualizar Usuario</p> <p><b>Actualizar usuario</b></p> <p>Usuario <input type="text" value="jjgarcia"/> Clasificación <input type="text" value="—Seleccione—"/></p> <p style="text-align: center;"><input type="button" value="actualizar"/></p>													
<b>Flujos Alternativos de Eventos</b>													
Acción del Actor	Respuesta del Sistema												
2. El administrador oprime “Cancelar”.	2.1 El sistema se mantiene en la misma interfaz y finaliza el caso de uso.												

Tabla 12 Descripción textual caso de uso “Gestionar Roles de Usuario”.

<b>Caso de Uso:</b>	Autenticar Usuario	
<b>Actores:</b>	Usuario	
<b>Resumen:</b>	El caso de uso se inicia cuando el Usuario introduce en el sistema su usuario y contraseña, y se le da acceso en dependencia del rol que desempeñe.	
<b>Precondiciones:</b>	Debe tener una cuenta en el dominio UCI.	
<b>Referencias:</b>		
<b>Prioridad:</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El Usuario introduce su usuario y su contraseña del dominio UCI y presiona el botón "Entrar".	1.1 El sistema verifica que los datos sean correctos determinando que permisos conceder de acuerdo al rol que representa el usuario. 1.2 El sistema muestra la interfaz correspondiente al nivel de acceso determinado, enviando a dicha interfaz los datos personalizados del usuario.	
<b>Flujos Alternativos de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	1.1 Si los datos introducidos por el usuario son incorrectos se muestra un mensaje de error. 1.2 Se retorna al paso 1.	
<b>Poscondiciones</b>	Debe quedar definido el nivel de acceso del Usuario.	

Tabla 13 Descripción textual caso de uso "Autenticar Usuario".

<b>Caso de Uso:</b>	Consultar Información	
<b>Actores:</b>	Usuario	
<b>Resumen:</b>	El caso de uso se inicia cuando el Usuario selecciona la opción consultar información.	
<b>Precondiciones:</b>	El usuario debe estar autenticado en el sistema.	
<b>Referencias:</b>		
<b>Prioridad:</b>	Opcional	
<b>Flujo Normal de Eventos</b>		
	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El Usuario selecciona la opción “consultar información”.	1.1 El sistema muestra una interfaz con las palabras almacenadas en la base de datos y su transcripción.
<b>Poscondiciones</b>	Debe quedar definido el nivel de acceso del Usuario.	

**Tabla 14 Descripción textual caso de uso “Consultar Información”.**

<b>Caso de Uso:</b>	Gestionar Puntos.
<b>Actores:</b>	Administrador de sistema y profesor.
<b>Resumen:</b>	El caso de uso consiste en que los profesores pueden gestionar los puntos de los estudiantes acumulados después de cada partida o sea modificar o eliminar. Pueden seleccionar cualquiera de estas opciones. Estos puntos quedan almacenados para que los estudiantes puedan verlos cuando deseen.
<b>Precondiciones:</b>	
<b>Referencias:</b>	
<b>Prioridad:</b>	Opcional

<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El actor solicita la opción de gestionar puntos.	1.1 Se muestra una página en la que aparecen las facultades registradas en el sistema.
1.2 El actor selecciona la facultad deseada.	1.3 Se muestran los grupos de esa facultad registrados en el sistema.
1.4 El actor selecciona el grupo deseado.	1.5 Se muestra un listado de los estudiantes del grupo con sus respectivos puntos. De cada estudiante se puede seleccionar la opción de eliminar o modificar sus puntos.
1.6 El actor selecciona la opción deseada.	1.7 Modificar (ir a sección modificar). Eliminar (ir a sección eliminar).
<b>Sección 2 “Eliminar puntos”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2. Se muestra al actor la opción de eliminar un estudiante o eliminarlos todos.
2.1. Selecciona una opción.	2.2. Se muestra un mensaje de que la eliminación se efectuó sin problemas.

Listado de estudiantes

### Listado de Puntos

Seleccione	Usuario	Puntos	Acciones
<input type="checkbox"/>	arcorrea	40	Elim   Act
<input type="checkbox"/>	ecarmenate	20	Elim   Act
<input type="checkbox"/>	jjgarcia	0	Elim   Act
<input type="checkbox"/>	yfperez	12	Elim   Act

Eliminar Marcados

### Sección 3 “Actualizar puntos”

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
	3. Se muestra un formulario el nombre del usuario y sus puntos en un cuadro de texto editable para que el actor pueda modificar su valor.
3.1 Modifica el valor de los puntos del estudiante.	3.2 Se almacenan los datos actualizados.

### Actualizar puntos de usuario

ecarmenate 20

Actualizar Nota

#### Flujos Alternativos de Eventos

Acción del Actor	Respuesta del Sistema
3.1. El actor oprime "Cancelar".	3.2. El sistema se mantiene en la misma interfaz y finaliza el caso de uso.
Poscondiciones	Se almacenan los datos insertados.

Tabla 15 Descripción textual caso de uso "Gestionar Puntos".

<b>Caso de Uso:</b>	Consultar Puntos	
<b>Actores:</b>	Usuario	
<b>Resumen:</b>	El caso de uso se inicia cuando el Usuario selecciona la opción consultar puntos.	
<b>Precondiciones:</b>	El usuario debe estar autenticado en el sistema.	
<b>Referencias:</b>		
<b>Prioridad:</b>	Opcional	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Usuario selecciona la opción "consultar puntos".	1.1 El sistema muestra una interfaz con los usuarios registrados en el sistema y sus puntos acumulados.	
<b>Poscondiciones</b>		

Tabla 16 Descripción textual caso de uso "Consultar Puntos".

## GLOSARIO DE TÉRMINOS

1. UCI: Universidad de las Ciencias Informáticas.
2. PHP: (acrónimo de "PHP: Hypertext Preprocessor") Procesador de hipertexto. Es un lenguaje para la programación Web.
3. BBC: La British Broadcasting Corporation (Corporación Británica de Difusión), más conocida como BBC, es la primera empresa de televisión y radio del Reino Unido. También es la cadena más grande de transmisión de noticias en el mundo. Produce servicios de programas e información, transmitiendo en la televisión, en la radio, y en Internet; siendo una de las cadenas de noticias más respetadas del mundo.
4. XHTML: acrónimo en inglés de eXtensible **H**ypertext **M**arkup **L**anguage (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web.
5. Cambridge University Press (conocido en inglés coloquialmente como CUP): es una editorial que recibió su Royal Charter de la mano de Enrique VIII en el año 1534, y es considerada una de las dos editoriales privilegiadas de Inglaterra
6. Fonética: Conjunto y estudio de los sonidos de un idioma.
7. XML: siglas en inglés de *Extensible Markup Language* (lenguaje de marcas ), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium. Permite definir la gramática de lenguajes específicos. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.
8. JSON: acrónimo de "**J**ava**S**cript **O**bject **N**otation", es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.
9. Software: La palabra «software» se refiere al **equipamiento lógico** o **soporte lógico** de un computador digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware).
10. Sparks: Es una librería que nos permite solventar dos deficiencias con las que nos podemos encontrar a la hora de afrontar un proyecto basado en CodeIgniter.

Mejora la caché que únicamente permite cachear páginas completas.  
Podremos incluir "vistas" dentro de "vistas".