



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



KLEARapp

Aplicación web para soporte de un sistema de escritura y generación de estructuras de conocimiento

MEMORIA DE PROYECTO PRESENTADA A LA FACULTAD DE
INGENIERÍA DE LA UNIVERSIDAD DE LA REPÚBLICA POR

Ma. Natalia Almeyda, Camila Deus, Alejandra Rivas

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS
PARA LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ELECTRICISTA.

TUTOR

Dr. Ing. Víctor González Universidad de la República

TRIBUNAL

Ing. Federico Davoine Universidad de la República
Ing. Claudina Rattaro Universidad de la República
Ing. Martín Rocamora Universidad de la República

Montevideo
martes 2 diciembre, 2014

KLEARapp

Aplicación web para soporte de un sistema de escritura y generación de estructuras de conocimiento, Ma. Natalia Almeyda, Camila Deus, Alejandra Rivas.

Esta tesis fue preparada en \LaTeX usando la clase iietesis (v1.1).

Contiene un total de 143 páginas.

Compilada el martes 2 diciembre, 2014.

<http://iie.fing.edu.uy/>

Agradecimientos

A todas y cada una de las personas que colaboraron, en forma directa o indirecta, en la realización de este proyecto.

Queremos destacar el interés y apoyo brindado por el tutor Dr. Ing. Víctor González Barbone a lo largo del proyecto.

También deseamos agradecer a nuestras familias ya que su apoyo fue fundamental en este camino. En particular a Rosa, Rosario, Francisco, Verónica, Stella, Héctor, Estelita y Gustavo.

Resumen

El proyecto KLEARapp propone la construcción de una aplicación web con las características y calidad de una aplicación para usuario final con objetivo de ser utilizada con fines educativos. Se pretende generar a través del mismo una herramienta que ayude a evaluar el proceso de aprendizaje de los estudiantes, apoyados en la infraestructura tecnológica que permite el desarrollo de esta modalidad.

En el presente documento se describen los elementos que constituyen la solución propuesta. Se detalla el origen del proyecto, así como los objetivos buscados y su alcance. Se presentan los programas utilizados, lenguaje de programación y framework (entorno de trabajo) que fueron soporte para la elaboración de esta aplicación. Se expone el resultado de un cuidadoso análisis de requerimientos planteados para el desarrollo del sistema, que dan lugar a los casos de uso. Se da un panorama general del desarrollo de las funciones, variables y demás componentes generadas para la implementación de esta aplicación, así como de los procedimientos para la utilización de la misma. Se detallan los problemas enfrentados a lo largo del proyecto, se fijan pautas de como seguir avanzando una vez terminado este proyecto y se realiza un análisis general del resultado obtenido.

Palabras claves: casos de uso, framework, interfaz, usuario.

Tabla de contenidos

Agradecimientos	I
Resumen	III
1. Introducción	1
1.1. Objetivo.	1
1.2. El Proyecto KLEAR.	1
1.3. Descripción del proyecto KLEARapp.	4
1.4. Estructura del documento.	5
2. Entorno tecnológico	7
2.1. Introducción.	7
2.2. Lenguaje de programación.	7
2.3. Framework para aplicaciones web.	8
2.4. Elección del motor de bases de datos.	10
2.5. Selección final.	10
3. Análisis de requerimientos	11
3.1. Introducción.	11
3.2. Requerimientos del sistema.	11
3.3. Análisis de los requerimientos.	14
3.3.1. Diagramas UML.	14
3.3.2. Casos de uso del sistema KLEARapp.	17
4. Diseño del sistema	21
4.1. Introducción.	21
4.2. La arquitectura de Software.	21
4.3. Patrón arquitectónico MVC.	21
4.4. Modelo MVC de la aplicación KLEARapp.	23
4.5. Componentes del sistema KLEARapp.	26
5. Modelo de base de datos.	31
5.1. Introducción.	31
5.2. Modelo de base de datos.	31
5.3. Modelo relacional.	32
5.4. Esquema de una base de datos relacional.	33

Tabla de contenidos

6. Implementación	35
6.1. Introducción.	35
6.2. Autenticación.	35
6.3. Matriculación.	36
6.4. Integración del sistema KLEAR a la aplicación KLEARapp.	37
6.5. Ingreso de texto y generación de grafo.	37
6.6. Asignación de pesos	38
6.7. Mostrar - ocultar y congelar - descongelar	38
6.8. Construcción de la evaluación.	39
6.9. Testing.	41
6.9.1. Unit testing.	42
6.9.2. Protocolo de testing.	43
7. Despliegue	47
7.1. Introducción.	47
7.2. Programas utilizados.	47
7.3. Detalle del paso a paso.	48
8. Evaluación	51
8.1. Introducción.	51
8.2. Evaluación a partir del Análisis de los casos de uso.	51
8.3. Evaluación a partir de los criterios de éxitos definidos.	52
9. Conclusiones y trabajo futuro.	55
9.1. Conclusiones.	55
9.2. Trabajo futuro.	56
Apéndices	57
A. Gestión de Proyecto	57
A.1. Introducción	57
A.2. Plan del proyecto KLEARapp.	57
A.2.1. Resumen.	57
A.2.2. Descripción del proyecto.	58
A.2.3. Objetivo general.	58
A.2.4. Alcance.	58
A.2.5. Criterios de éxito.	58
A.2.6. Actores.	58
A.2.7. Supuestos.	59
A.2.8. Restricciones.	59
A.2.9. Especificación funcional del proyecto.	59
A.2.10. Objetivos específicos.	60
A.2.11. Entregables asociados.	60
A.2.12. Tareas.	61
A.2.13. Cronograma detallado del proyecto.	63
A.2.14. Análisis de costos.	63

A.2.15. Análisis de Riesgos.	63
A.3. Análisis de la gestión realizada.	66
B. Análisis de casos de uso por áreas	69
B.1. Administración de usuarios	69
B.1.1. Registrarse en la aplicación Klearapp con rol alumno.	69
B.1.2. Registrarse en la aplicación Klearapp con rol docente.	71
B.1.3. Registrarse en la aplicación Klearapp con rol administrador.	72
B.1.4. Solicitar matriculación a un curso.	74
B.1.5. Confirmar matriculación de Alumno.	75
B.1.6. Eliminar usuario Alumno de un curso.	77
B.1.7. Eliminación de usuario Alumno por parte de un docente.	78
B.1.8. Deshacer eliminación de usuario Alumno por parte de un docente.	79
B.1.9. Solicitar matriculación con rol docente en curso existente.	80
B.1.10. Confirmar matriculación de Docente.	82
B.1.11. Eliminar usuario docente de un curso.	84
B.1.12. Restablecer usuario docente a un curso por parte de otro docente del curso.	85
B.1.13. Restablecer matriculación de usuario docente a un curso por parte del administrador.	86
B.1.14. Eliminación de usuario docente de todos los cursos por parte del administrador.	88
B.2. Administración de Cursos	89
B.2.1. Crear un nuevo curso en el sistema KLEARapp.	89
B.2.2. Borrar un curso del sistema KLEARapp por parte de un docente.	90
B.2.3. Modificar curso.	91
B.2.4. Borrar un curso del sistema KLEARapp por parte del administrador.	92
B.2.5. Usuario ingresa al curso.	94
B.3. Administración de Unidad de Aprendizaje (U.A.)	95
B.3.1. Crear una U.A.	95
B.3.2. Eliminar una U.A.	96
B.3.3. Modificar una U.A.	97
B.4. Manejo de tareas	99
B.4.1. Crear una Tarea.	99
B.4.2. Modificar una Tarea.	100
B.4.3. Hacer visible/invisible una tarea.	100
B.5. Manejo del texto de referencia	102
B.5.1. Editar texto de referencia.	102
B.5.2. Descongelar texto de referencia.	104
B.5.3. Congelar texto de referencia.	105
B.5.4. Generar grafo del texto.	107
B.5.5. Asignar pesos a nodos y arcos.	107

Tabla de contenidos

B.5.6. Asignar pesos a nodos de forma particular.	108
B.5.7. Modificar pesos a nodos y arcos.	109
B.6. Textos borradores	111
B.6.1. Crear texto borrador.	111
B.6.2. Modificar texto borrador.	112
B.7. Léxico y reglas.	113
B.7.1. Administrar léxicos en la aplicación.	113
B.7.2. Corregir léxico y sintaxis en un texto (texto de referencia o borrador).	114
B.8. Administración de evaluación	115
B.8.1. Crear evaluación.	115
B.8.2. Modificar evaluación.	116
B.8.3. Hacer visible/invisible una evaluación.	117
B.8.4. Realizar la evaluación.	118
B.8.5. Docente accede a las calificaciones de la evaluación asociada a una U.A.	120
B.8.6. Docente accede a la evaluación de un alumno.	120
B.8.7. Alumno accede a la calificación que obtuvo en una evaluación.	122
B.8.8. Alumno accede a una evaluación que realizó.	122
B.9. Suministro de ayuda	123
B.9.1. Solicitar ayuda haciendo clic en el ícono “?” de la ventana.	123
B.9.2. Solicitar ayuda haciendo clic en la palabra “Ayuda” del menú lateral.	124
C. Manuales de usuario	127
C.1. Introducción.	127
Referencias	129
Índice de tablas	131
Índice de figuras	132

Capítulo 1

Introducción

El avance vertiginoso de los últimos años en las tecnologías de la información y la comunicación (TICs) han generado un impacto en el ámbito social, económico, cultural y educativo. En particular en el ámbito educativo existe una relación muy estrecha con el avance tecnológico que se potencia sistemáticamente. Esto hace que los modelos de aprendizaje cambien y aumenten sus posibilidades adaptándose a las nuevas necesidades de las instituciones y usuarios. [1]

Estamos viviendo una era donde la optimización del tiempo, la automatización y el fácil acceso se han vuelto características claves para toda actividad humana. En este contexto el proyecto KLEARapp adquiere total vigencia.

1.1. Objetivo.

El proyecto KLEARapp tiene como objetivo desarrollar una aplicación web para soporte del sistema KLEAR (Knowledge and Learning for Education, Assessment and Research) [2] que permite la corrección automática de texto bajo una estructura específica.

Esta aplicación web debe contar con las funcionalidades necesarias para acceder al sistema KLEAR desde un navegador web, con las características y calidad de una aplicación para usuario final. La presentación de la aplicación, debe tener una interfaz de usuario clara, sencilla de manejar, eficiente y grata para el uso por estudiantes y docentes.

1.2. El Proyecto KLEAR.

KLEAR es un proyecto de investigación elaborado en la tesis de doctorado del Dr. Ing. Víctor González, que propone el uso de un sublenguaje para la construcción de representaciones de conocimiento en forma de grafos semánticos, destinado a la mejora de la escritura y la evaluación de respuestas abiertas (a texto libre).

Capítulo 1. Introducción

La ejecución del sistema KLEAR, el cual está escrito en lenguaje Python, permite cumplir las siguientes etapas:

1. Se ingresa un texto que se denomina texto de referencia, escrito en sublenguaje. Este sublenguaje es un subconjunto del lenguaje natural, conformado por términos particulares referidos a un área de conocimiento y ciertas reglas de producción que permitirán su procesamiento en el sistema para posibilitar la extracción de conocimiento.

El sublenguaje desarrollado en el proyecto KLEAR está en inglés porque son mayores los recursos lingüísticos en este idioma, que para el castellano. [2]

Todos los textos ingresados deben ser escritos en este sublenguaje.

El siguiente es un ejemplo de texto escrito en sublenguaje extraído de [2]:

```
Venus moved [across the face [of the Sun]].
the exact times varied [with the position [of the observer]] [by
a maximum [of eight minutes]].
the transits [of Venus] are very rare.
the transits [of Venus] are a predictable celestial phenomenon
```

2. El texto pasa por una validación. Se aplican reglas que definen si tiene la estructura sintáctica correcta aceptada por el sublenguaje. De no cumplir con estas reglas de producción se indica al usuario que debe realizar las correcciones pertinentes para ser validado.
3. Luego se genera un árbol semántico del texto que es la base para la generación del grafo semántico. En este se muestra la relación entre los componentes sintácticos del texto.
4. Se genera el grafo semántico, el cual es una representación gráfica (imagen en formato PNG) de los contenidos impartidos en el texto ingresado. Este grafo está compuesto por distintas figuras geométricas en colores dependiendo de la categoría léxica que tienen las palabras escritas, es decir se diferencian claramente cuáles son los sustantivos, verbos, adjetivos, etc.

La figura 1.1 muestra el grafo generado por el texto de referencia.

1.2. El Proyecto KLEAR.

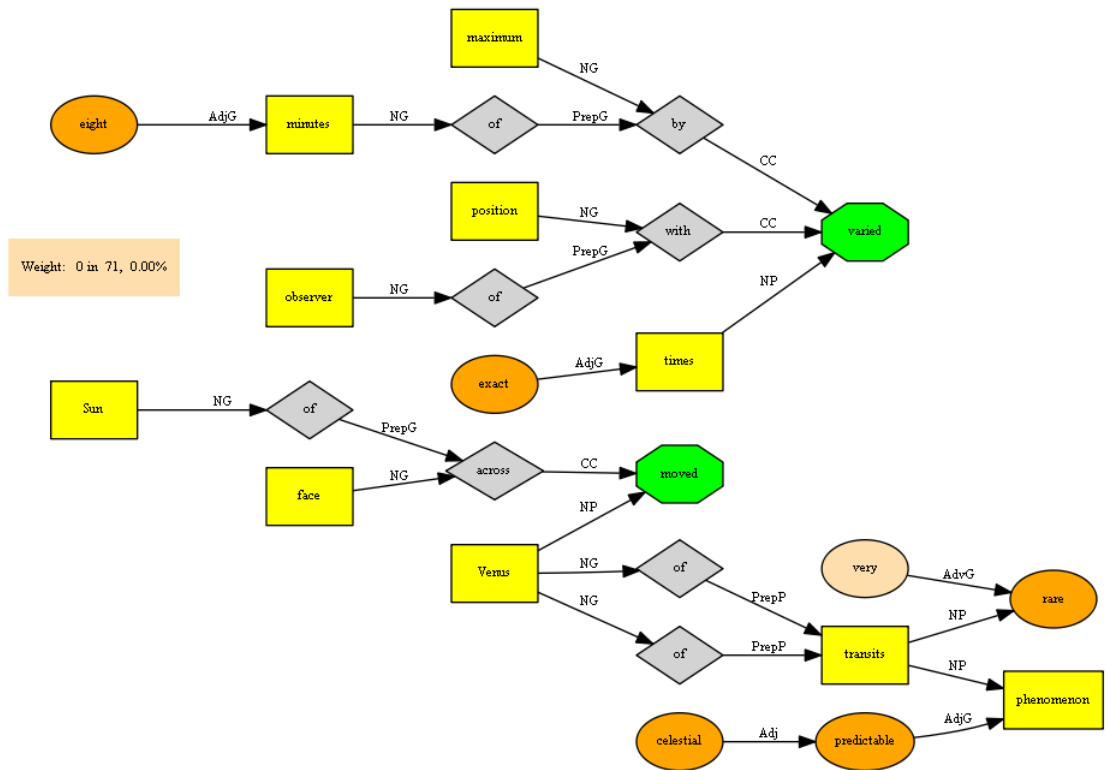


Figura 1.1: Grafo del texto de referencia generado por el sistema KLEAR

5. Existe una segunda instancia de ingreso de texto, que puede llamarse texto respuesta, este debe estar contenido en el texto de referencia. El mismo es validado de igual forma que el de referencia. De esta forma se genera el grafo respuesta.

Ejemplo de un posible texto respuesta:

the times varied [with the position [of the observer]].
the transits [of Venus] are very rare.

6. Se realiza una comparación entre el grafo de referencia (generado por el texto referencia) y el grafo respuesta. En esta instancia se genera una nueva representación gráfica donde se pueden ver los conceptos reconocidos en el grafo referencia en color y los que no, en tono gris, mostrando de forma inmediata la calificación obtenida. Para realizar la calificación es necesario que previamente se hayan asignado pesos según su categoría léxica. Se le llama peso a un valor numérico que se asigna a las palabras que componen el grafo.

La figura 1.2 muestra el resultado de la comparación, con la calificación obtenida:

Capítulo 1. Introducción

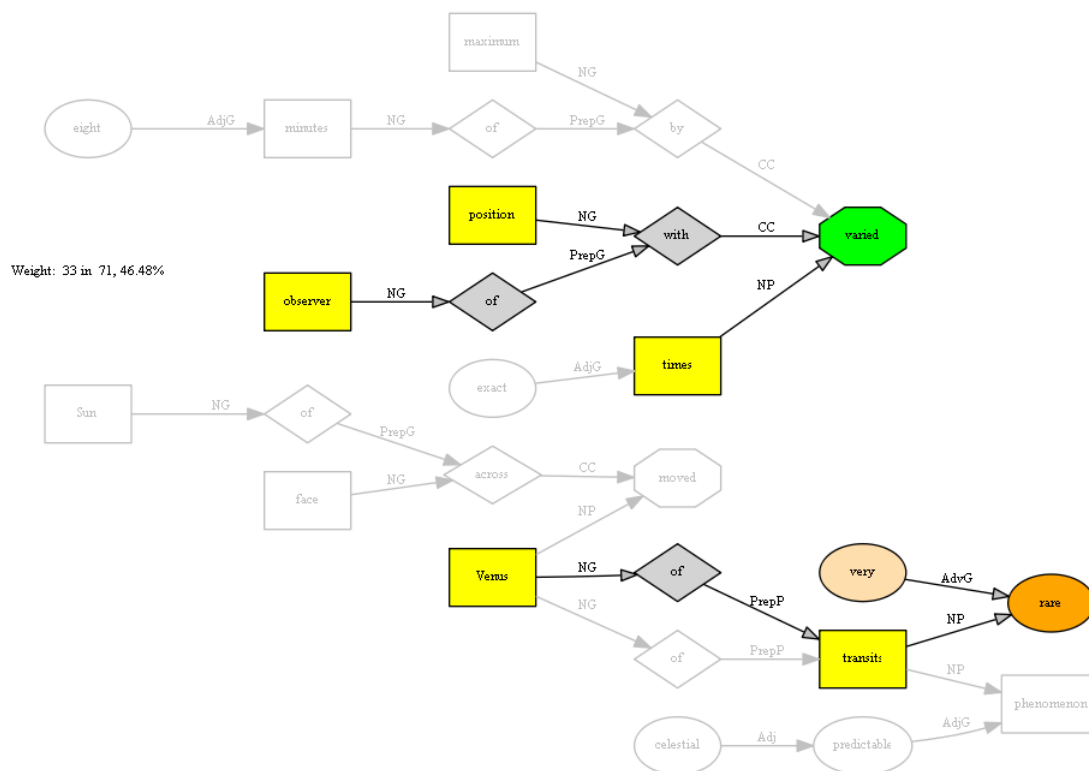


Figura 1.2: Grafo resultante de la comparación con el texto respuesta generado por el sistema KLEAR.

Este mecanismo se plantea con el fin de ser desarrollado para trabajar en una clase, de manera que se escriba el material de estudio de una unidad didáctica, en la que los alumnos son quienes con supervisión y ayuda del docente van creando de manera conjunta el texto de referencia de esa unidad de aprendizaje. A partir de este texto de referencia, se obtiene el grafo de referencia con el cual se compararán las respuestas dadas por los estudiantes en la instancia de evaluación.

Para ello es imperioso que el docente realice el cuestionario con preguntas cuyas respuestas estén contenidas en el texto de referencia que se realizó en clase. Realizando la comparación entre el texto respuesta, escrito de igual forma, con el mismo sublenguaje y reglas que el de referencia, el estudiante puede ver el reconocimiento de su respuesta y la nota lograda.

1.3. Descripción del proyecto KLEARapp.

La representación de conocimiento ofrece interesantes posibilidades para enseñanza y aprendizaje, obligando al estudiante a seleccionar, jerarquizar y relacionar conceptos. Esto es lo que se hace habitualmente en la creación de mapas

1.4. Estructura del documento.

conceptuales. El uso del lenguaje escrito ha sido y es una forma tradicional de expresar la comprensión de un tema, a través de resúmenes o ensayos cortos.

Este proyecto propone la construcción de una aplicación web apta para el uso por estudiantes y docentes, que implemente la construcción de grafos semánticos a partir de textos usando los desarrollos del proyecto KLEAR. Esta aplicación, con las facilidades adecuadas para el uso por estudiantes, es un elemento fundamental para poder realizar pruebas del sistema KLEAR, y avanzar en la investigación.

El sistema KLEARapp se desarrolla en lenguaje de programación Python, empleando el entorno de trabajo web2py basado en este lenguaje. En el siguiente capítulo se ampliará sobre Python y web2py.

El modelo organizativo empleado para el diseño del software es Model View Controller (MVC).

La presentación de la aplicación, tiene como premisa tener una interfaz de usuario clara, sencilla de manejar, eficiente y grata para el uso por estudiantes y docentes.

1.4. Estructura del documento.

Se presenta a continuación un panorama general de los capítulos que componen este documento.

En el capítulo 2 se describen las tecnologías relacionadas con la aplicación y su vínculo con la solución propuesta.

En el capítulo 3 se presenta el estudio de la solución a partir de los requerimientos establecidos para la implementación del sistema. Se listan las funcionalidades y atributos con los que se debe cumplir.

En el capítulo 4 se realiza el diseño del sistema, se definen los módulos y sus interacciones. Se presenta el diagrama de la arquitectura.

En el capítulo 5 se describe el significado de los modelos de bases de datos y cuál es el esquema del modelo relacional que se define para el diseño.

En el capítulo 6 se describe la codificación implementada para resolver aspectos significativos de la aplicación y se presenta un protocolo de testeo para los casos de uso.

En el capítulo 7 se detallan los pasos a seguir para la instalación de este software.

Capítulo 1. Introducción

El capítulo 8 presenta la evaluación del proyecto analizando en qué medida el Análisis de los casos refleja al producto final y verificar el cumplimiento de los criterios de éxito.

Conclusiones y trabajos futuros se describen en el capítulo 9.

Al final del documento se presentan los siguientes Apéndices:

El Apéndice A presenta un resumen de lo planificado al inicio de este proyecto, y se analiza el grado de cumplimiento.

El Apéndice B presenta la descripción completa de los casos de uso.

Finalmente en el Apéndice C se encuentran los manuales de uso, correspondientes a alumnos, docentes y administradores

Capítulo 2

Entorno tecnológico

2.1. Introducción.

En este capítulo se describen las tecnologías disponibles para desarrollar la aplicación y la elección realizada en la solución propuesta.

2.2. Lenguaje de programación.

El sistema KLEAR como se explica en el capítulo anterior, se desarrolló en el lenguaje de programación Python. Esto motivó a que fuera el lenguaje elegido para el desarrollo de la aplicación KLEARapp, sin necesidad de convertir los módulos de KLEAR a otro lenguaje de programación, facilitando el trabajo.

Python posee una licencia de código abierto denominada Python Software Foundation License (PSFL por sus siglas en inglés) que fue declarada por la Open Source Initiative (OSI) licencia de software libre por cumplir con los requisitos que ésta establece para ello [3].

Ser de software libre implica que se tiene libertad para estudiarlo y adaptarlo a las necesidades particulares, permitiendo también el desarrollo de nuevos productos o la realización de mejoras. Esto exige el acceso al código fuente.

Además, la licencia permite la libertad de uso y redistribución; es posible la instalación del software tantas veces y en tantas máquinas como sea deseable.

El software libre se puede vender si se desea, pero como cualquiera puede redistribuirlo sin pedir un pago por ello, se hace prácticamente imposible obtener dinero a partir de su distribución. [4]

Soporta distintos paradigmas de programación: orientado a objetos, funcional e imperativo. Esto le permite al desarrollador tener mayor libertad en el momento

Capítulo 2. Entorno tecnológico

de pensar el modelo de solución.

Además cuenta con tipado dinámico. Esto refiere a que el propio programa asigna el tipo de variable en tiempo de ejecución, interpretando el valor que toma cada variable. Una vez que el lenguaje determina el tipo de variable solo se puede modificar el mismo en forma explícita, lo que se conoce como ‘fuertemente tipado’.

Otra de las ventajas de utilizar Python es que es multiplataforma, es decir que puede ser utilizado en distintos sistemas operativos sin grandes cambios [5].

Python incluye una gran cantidad de librerías estándar, lo que le permite al desarrollador utilizarlas sin necesidad de descargarlas, simplemente importándolas. También permite incorporar otros paquetes, en este proyecto se incluyó el The Natural Language Toolkit (NLTK), que está formado por un conjunto de bibliotecas y programas utilizados para el procesamiento del lenguaje natural [6], indispensable para la ejecución de los módulos del sistema KLEAR. Sobre la instalación de este paquete ver Capítulo 7.

El aprendizaje de dicho lenguaje resultó fundamental para entender el sistema KLEAR e integrarlo a la aplicación creada.

2.3. Framework para aplicaciones web.

Partiendo de la base que no había experiencia previa en el desarrollo de aplicaciones web, se consideró como mejor alternativa la utilización de un framework, un entorno de trabajo que facilita considerablemente el desarrollo de aplicaciones web con calidad de producción. Permite crear una aplicación desde cero a partir de una base que organiza y une los diferentes componentes del proyecto.

Algunos de los frameworks más conocidos son: Ruby on Rails, CodeIgniter, Kohana, Django, CakePHP, Senz Framework, Yii, Pylons, Catalyst, Symfony, web2py.

Teniendo en cuenta los motivos expuestos al principio del capítulo, que determinaron que Python era el lenguaje de programación apropiado para desarrollar la aplicación y que se buscaba una solución de software libre, se eligieron como posibles frameworks aquellos que estaban programados en base a Python y con licencia gratuita, por lo que quedaron como opciones: Django, Pylons, web2py [7]. Por último se seleccionó a web2py como el framework a utilizar para el desarrollo de KLEARapp.

Web2py es un framework con varias características interesantes a la hora de utilizarlo para el desarrollo de una aplicación web, en particular se adapta muy bien a las necesidades del desarrollo de la aplicación en cuestión. Es un software libre que se puede descargar desde su página oficial **www.web2py.com**. Es liviano,

2.3. Framework para aplicaciones web.

pesa 1,4 MB incluyendo sus componentes básicos. Utiliza el paradigma MVC, apropiado para aplicaciones donde existe una fuerte interacción con el usuario y hace factible el mantenimiento y la escalabilidad.

Otra de las características importantes es su alta seguridad. Según la Open Web Application Security Project, web2py cuenta con recursos para resolver los mayores conflictos de seguridad que puede presentar una aplicación web. Por más información acerca de cómo resuelve web2py el tema de seguridad, visitar la página <http://web2py.com/books/default/chapter/36/01/introduccion>.

Es reconocido por su independencia en cuanto a la instalación de otras tecnologías. Se puede utilizar simplemente descargando el programa web2py, habiendo instalado previamente Python, las librerías estándar y en este caso particular (por los motivos que se explicaron más arriba) el paquete NLTK.

Su gran comunidad de desarrolladores permite que continuamente se avance en la herramienta. Es una comunidad activa, desde su creador que siempre está involucrado en las consultas y todos sus seguidores que aportan conocimiento y código para nuevos desarrollos. Acerca de las versiones, se destaca su cuidado para evitar incompatibilidades con versiones anteriores.

A diferencia de otros frameworks tanto los módulos como los controladores no son importados, sino que son ejecutados por el programa en el momento. Esto es una ventaja porque se puede instalar, modificar o desinstalar la aplicación sin la necesidad de reiniciar el servidor web.

Acerca de sus componentes, se puede decir que proporcionan un sistema sencillo de manejo de la base de datos. Utiliza una capa de abstracción (DAL), que permite generar un objeto de base de datos a partir de un objeto Python, sin la necesidad de escribir código en lenguaje de consulta estructurado (SQL por sus siglas en inglés). La DAL genera SQL en tiempo real. Web2py tiene incorporados distintos motores de base de datos como: `sqlite3`, `pymysql`, `pg8000` y `imaplib` [8]. Por defecto, el framework trae una plantilla de HTML. La misma consiste en una vista escrita en lenguaje HTML que se guarda con el nombre `layout.html`. Cada vista puede utilizar dicha plantilla, para lo cual simplemente debe agregar al comienzo del código la expresión `extend 'layout.html'`. De esta forma lo que muestra la interfaz se presenta con el formato establecido en el archivo.

Web2py también incluye hojas de estilo, que contienen un conjunto de instrucciones que se ocupan de dar formato a la presentación: tipo de fuente, colores, sombreados, etc. En particular, el framework trabaja con el lenguaje de hoja de estilo en cascada (en inglés Cascading style sheet (CSS)) que permite presentar un documento estructurado escrito tanto en HTML como en XML.

Considerando estas ventajas se eligió el uso de este framework.

2.4. Elección del motor de bases de datos.

En el momento de elegir el motor de bases de datos se tuvo en cuenta que el framework Web2py trae por defecto el sistema de gestión de las bases de datos SQLite. El mismo responde al esquema referencial y por sus características se puede concluir que se adapta a las exigencias establecidas, por lo que resulta una buena elección.

Sin embargo, es importante aclarar que se podría utilizar otros tipos de sistemas de gestión de bases de datos, simplemente se debe ejecutar desde la distribución fuente e instalar el controlador (drive) adecuado para el motor que se desea incorporar. Una vez hecho esto, se inicia Web2py desde la fuente y el mismo lo encuentra.

2.5. Selección final.

Luego de investigar y analizar las diferentes herramientas para llevar a cabo la implementación de la aplicación se optó por emplear las que se detallan a continuación:

- Python.
- web2py.
- bibliotecas NLTK.
- sqlite3 como motor de base de datos.
- bibliotecas de KLEAR.

Sobre la instalación de los programas Python, web2py y bibliotecas NLTK ver Capítulo 7.

Capítulo 3

Análisis de requerimientos

3.1. Introducción.

Para crear una aplicación de software es necesario describir el problema y las necesidades, en qué consisten y qué debe hacerse. En este capítulo se presenta el estudio de la solución a partir de los requerimientos establecidos para la implementación del sistema. Se listan además las funcionalidades y atributos con los que se debe cumplir.

3.2. Requerimientos del sistema.

Los requerimientos son parte de la descripción de las necesidades o deseos que debe cumplir la aplicación KLEARapp. Una adecuada especificación de los mismos es fundamental para asegurar el éxito de la aplicación y así reflejar de forma inequívoca lo que se quiere del producto. Un buen planteo de la lista de requerimientos reduce los riesgos de alejarse de las expectativas que tiene el cliente. Esta lista es de utilidad para verificar, una vez terminada la aplicación, si se ha logrado lo que el cliente solicitó.

Para elaborar la lista de requerimientos se identificaron los usuarios, los procesos y recursos involucrados y las relaciones existentes entre ellos. A partir de esta información se definen los límites del sistema y el alcance de las tareas requeridas.

A continuación se detallan las facilidades a proveer al usuario, obtenidas a partir de la entrevista entre el cliente y los desarrolladores, donde se identifican las entidades de la aplicación, los roles de los usuarios y funcionalidades.

Identificación de entidades:

- Unidad de Aprendizaje: La unidad de aprendizaje reúne el conocimiento necesario para cumplir un objetivo de aprendizaje. Consiste en:

Capítulo 3. Análisis de requerimientos

- Título.
 - Subtema, tema, asignatura a la que corresponde.
 - Objetivos de aprendizaje.
 - Descripción.
 - Textos (uno o más); los grafos se generan a partir de los textos, los textos son producidos por los alumnos.
 - Asignación de pesos a nodos y arcos (una vez “congelados” los textos).
 - Preguntas de evaluación.
 - Respuestas de los alumnos a las preguntas de evaluación (texto y grafo).
 - Resultados de evaluación, por alumno.
- Esquema Sublenguaje y Representación de Conocimiento:
 - Sublenguaje: reglas de producción, léxico.
 - Reglas de conversión (gramática de dependencia) hacia el árbol de dependencias.
 - Reglas de conversión del árbol de dependencias a grafo.

Estos elementos son módulos (archivos) que se cargan externamente, permiten cambiar el esquema de sublenguaje y representación de conocimiento usados. También debe ser posible complementar el esquema básico con:

- Léxico complementario: nuevas palabras.
- Reconocimiento de estructuras complementarias: ej. cómo tratar medidas, o fechas.

Identificación de roles:

Se reconocen estos roles, con diferentes funcionalidades requeridas:

- Docente:
 - Administración de actividades de aprendizaje.
 - Crea y administra tareas.
 - Puede mantener las tareas invisibles a los estudiantes, mientras las prepara, luego hacerlas visibles.
 - Puede editar textos de referencia asociados a una tarea.
 - Asigna pesos a los nodos y arcos de los grafos, para la calificación.
 - “Congela” la tarea, impidiendo que los estudiantes sigan modificando los textos de referencia. También puede descongelarla, si fuera necesario.

3.2. Requerimientos del sistema.

- Propone preguntas de evaluación, ocultas a los estudiantes.
- Fija período de visibilidad de las preguntas, para la instancia de evaluación.
- Examina sin modificar las respuestas de los estudiantes; ve textos, grafos y calificación.
- Obtiene una lista de calificaciones de los estudiantes, exportable (ej. planilla electrónica).
- Administración de cursos.
 - Administra los cursos a su cargo: crea, modifica, borra, respalda o exporta.
- Administración de alumnos.
 - Administra usuarios en rol de alumnos.
- Administración de léxico y reglas.
 - Complementa el léxico o las reglas.
 - Documenta los cambios; los hace accesibles a través de la ayuda que proporciona el sistema.
- Alumno:
 - Accede a la tarea cuando es liberada.
 - Crea textos de referencia, visualiza grafos.
 - Complementa el léxico.
 - Consulta la ayuda sobre el uso del sistema, sublenguaje, conversión a grafo, etc.
 - Los textos y grafos de referencia para la tarea se construyen colaborativamente, diferentes alumnos pueden alterar la misma estructura.
 - Dispone de un “borrador” donde puede crear textos y grafos no vinculados a ninguna tarea.
- Administrador:
 - Instala el sistema.
 - Configura el sistema, cambia módulos de sublenguaje y representación.
 - Administra usuarios en rol Docente
 - Puede borrar cursos.

Uso y funcionalidad:

- El Editor:
 - Provee ayuda léxica, informa si una palabra no está en el léxico, si hay sinónimos canónicos, etc.

Capítulo 3. Análisis de requerimientos

- Provee ayuda sintáctica, muestra árboles parciales cuando una oración no es validada por la gramática.
- Conserva lo ya escrito, permite corregir sin reingresar todo.
- Niveles de ayuda:
 - Uso general del sistema.
 - Ayuda del esquema de sublenguaje y representación en uso.

3.3. Análisis de los requerimientos.

En este apartado se analizan los requerimientos, elaborándose un modelo de procesos implicados que refleje las principales entidades intervinientes, las relaciones entre ellas y con el sistema que compone la aplicación KLEARapp.

El estudio de los requerimientos es el punto de partida para comenzar a entender qué se pretende lograr del desarrollo de este sistema.

3.3.1. Diagramas UML.

El lenguaje Unificado para la Construcción de Modelos, UML, permite modelar al sistema, dar una visión global de los diferentes modelos que componen la aplicación y sus interacciones con los diferentes actores.

A continuación se muestran los diagramas UML:

- Identificación de Roles:

La figura 3.1 presenta el rol de los distintos actores que interactúan con el sistema que se quiere diseñar.

3.3. Análisis de los requerimientos.

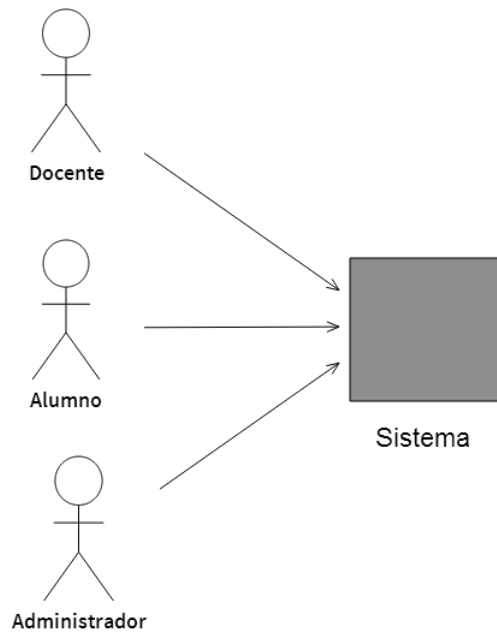


Figura 3.1: Identificación de Roles.

- Docente: En la figura 3.2 se muestra las funcionalidades que surgen de los requerimientos y están encomendadas a los usuarios con rol docente.

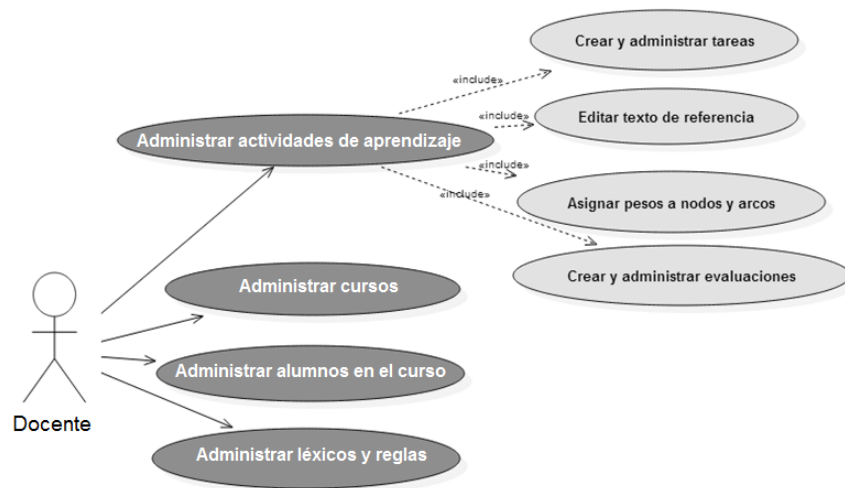


Figura 3.2: Funcionalidades referentes al rol Docente.

- Alumno:
En la figura 3.3 se presentan las funcionalidades destinadas a los usuarios

Capítulo 3. Análisis de requerimientos

con rol alumno.

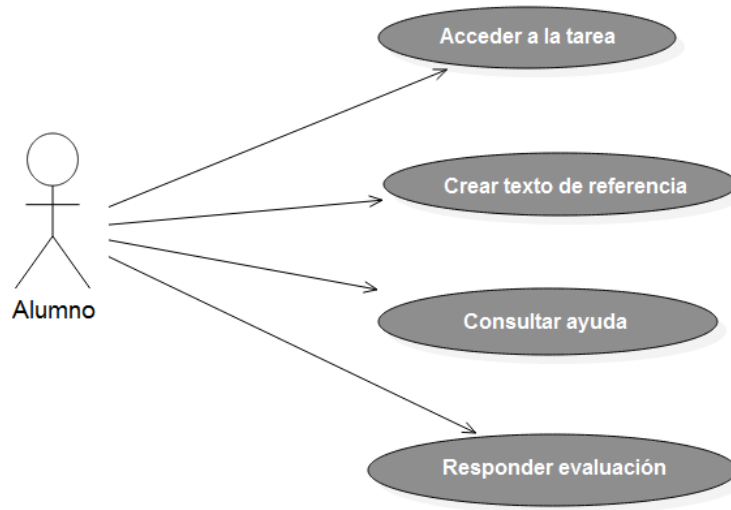


Figura 3.3: Funcionalidades referentes al rol Alumno.

- Administrador: En la figura 3.4 se presentan las tareas asignadas al rol administrador.

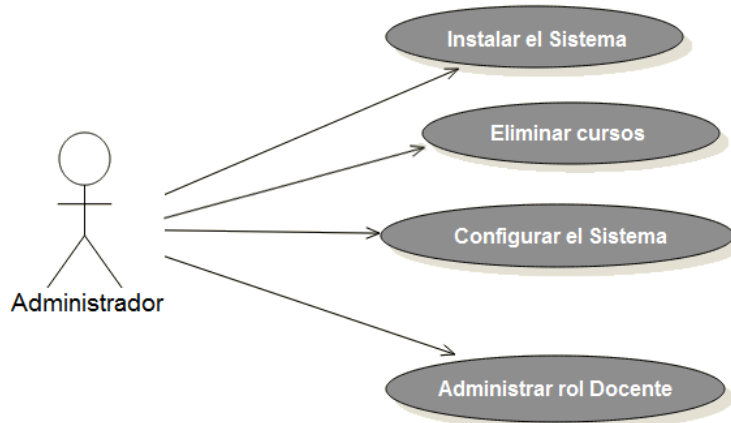


Figura 3.4: Funcionalidades referentes al rol Administrador.

3.3. Análisis de los requerimientos.

3.3.2. Casos de uso del sistema KLEARapp.

Para comprender cuáles deben ser las funcionalidades del sistema KLEARapp, se estudian los casos de uso.

A través de los casos de uso se describe la interacción entre los actores y el sistema. Estos son una descripción de un proceso relativamente largo de inicio a fin incluyendo varios pasos o transiciones [9].

Para realizar la identificación de los casos de uso se analizan los requerimientos especificados por el cliente, obteniendo una serie de eventos externos a los que el sistema debe responder. Los actores son quienes interactúan con el sistema generando estímulos de entrada y salida del sistema.

En el Apéndice B se encuentra un análisis detallado de los casos de uso pertinentes para la realización de esta aplicación web. Para iniciar el desarrollo de las funcionalidades que tiene KLEARapp se comienza con la administración de usuarios, que es lo primero con lo que se enfrenta el usuario.

Administración de usuarios.

Todos los usuarios que ingresan a la aplicación deben registrarse para poder acceder a los recursos que se ofrecen. Disponen así de un nombre de usuario que es su documento de identidad y una contraseña, que les permite acceder a las funcionalidades del mismo según su rol (alumno, docente o administrador). Una vez registrados, ya pueden iniciar sesión. A continuación los usuarios registrados pueden acceder a los distintos cursos.

Gestión de Cursos.

Los cursos dividen áreas temáticas de conocimiento.

En cada curso participan los docentes quienes son los encargados de la creación y modificación de los aspectos que definen a los mismos (título y descripción) a través de formularios de edición. La eliminación de un curso puede ser realizada por sus docentes o por el administrador a través de su interfaz.

Los docentes que crean cursos, automáticamente quedan matriculados en él. Alumnos y demás docentes deben solicitar matriculación para acceder a un curso particular. De esta forma se da la potestad al docente del curso de decidir quién ingresa por medio de un listado de solicitudes. Mientras están a la espera de la aceptación, las solicitudes quedan en estado pendiente. Una vez que se responde afirmativamente pueden acceder al curso. Todo usuario puede desmatricularse si así lo desea.

Capítulo 3. Análisis de requerimientos

En cada curso se crean unidades de aprendizaje las cuales refieren a un tema particular.

Administración de Unidades de Aprendizaje.

Una unidad de aprendizaje (U.A.) es un bloque que reúne todos los recursos (tareas, evaluaciones, textos de referencia) necesarios para cumplir con un objetivo de aprendizaje.

El docente es quien administra los bloques, tiene permiso para crear y quitar una o varias U.A.. A su vez puede agregar tareas y evaluaciones a los mismos.

Una vez que el docente crea una U.A. debe especificar una lista de campos (título, descripción, subtema, etc.) referidos al tema al que concierne el bloque.

Cada U.A. es visible también para los alumnos, quienes podrán acceder a las tareas y evaluaciones que se presentan.

Alumnos y docentes pueden crear borradores sobre cada tema tratado dentro de las U.As.

Manejo de Tareas y Borradores.

La tarea de una U.A. contiene el material necesario para alcanzar el objetivo de aprendizaje correspondiente a esa U.A.. Para esto se incluye un espacio donde se puede editar un texto de forma colaborativa por todos los alumnos del curso. A estos textos se los denominan texto de referencia, como figura más adelante.

Los docentes crean, modifican y eliminan las tareas dentro de sus cursos. Tienen la posibilidad de elaborarlas y dejarlas ocultas a los estudiantes hasta que lo consideren necesario.

Los borradores se presentan como otro espacio de edición donde los alumnos pueden practicar el desarrollo de los temas y luego copiar las ideas de allí para integrarlas al texto de referencia que se crea en la tarea.

Todo texto ingresado pasa por un proceso de validación tanto de léxico como de reglas, lo que permiten la construcción del grafo semántico asociado al texto.

Generación del Texto de Referencia.

Los textos de referencia están asociados a una tarea dada por el docente y contienen todos los conceptos e información correspondiente a esa tarea asignada.

Alumnos y docentes crean de forma colaborativa estos textos en sublenguaje mientras la tarea está visible y activa para editar. Una vez que un texto es finali-

3.3. Análisis de los requerimientos.

zado y el docente entiende que está completo, puede congelar la tarea impidiendo que los estudiantes sigan realizando modificaciones al mismo.

Estos textos serán luego la referencia que tienen los alumnos para estudiar en la instancia de evaluación. El docente debe asignar los pesos a nodos y arcos que conforman el grafo de referencia antes de la instancia de evaluación.

Asignación de pesos a nodos y arcos.

Una vez que el texto de referencia es finalizado, el docente genera el grafo de referencia y asigna pesos a nodos y arcos. Es decir, asigna un valor numérico a las palabras que componen el grafo, con lo cual se realizará la comparación en instancia de evaluación para obtener la calificación.

La asignación de estos pesos se puede realizar en dos etapas. Una primera, en la cual la asignación se realiza de acuerdo a las categorías léxicas que tienen las palabras del texto, y una segunda etapa opcional en la que el docente puede elegir en forma particular la palabra a la que le quiere asignar peso independientemente de su categoría.

Si lo desea puede verificar como quedó la asignación realizada, desplegándose un listado de cada palabra y peso asociado.

Cumplida esta instancia, están dadas las condiciones para la realización de la evaluación.

Realización de Evaluaciones.

La evaluación dentro de cada U.A. es creada por los docentes del curso. Deben proponer preguntas cuyas respuestas estén contenidas dentro del texto de referencia. Las preguntas pueden ser planificadas con anticipación y dar visibilidad a los estudiantes en el momento de ser realizada.

Los estudiantes hacen la prueba, contestando las preguntas en el mismo sublenguaje y con los conocimientos obtenidos en el proceso de generar el texto de referencia.

Inmediatamente de terminar la evaluación se les presenta la calificación obtenida en esa instancia.

En el Apéndice C, se encuentran los manuales que paso a paso muestran cómo ejecutar las funcionalidades anteriores.

Capítulo 4

Diseño del sistema

4.1. Introducción.

En el presente capítulo se lleva el análisis a un modelo de solución en forma abstracta, el cómo. Basándose en el análisis realizado en el capítulo anterior, se definen módulos y sus interacciones, y se presenta el Diagrama de Arquitectura y el Diseño Detallado.

4.2. La arquitectura de Software.

La arquitectura de software es la forma en que se organizan los componentes de un sistema, indicando funcionamiento e interacción entre los mismos.

Es el diseño de más alto nivel de la estructura del sistema. Se basa en el análisis de los requerimientos y en el planteo de los casos de uso.

El patrón arquitectónico dentro de la arquitectura de software, define la estructura básica del sistema y da una descripción de los elementos y el tipo de relación que tienen. Provee un conjunto de normas para su organización. [10] [11]

4.3. Patrón arquitectónico MVC.

Web2py está diseñado de forma de programar siguiendo las buenas prácticas de la ingeniería de software. Por este motivo utiliza el patrón arquitectónico Modelo - Vista - Controlador (MVC) para que el desarrollador trabaje con esta arquitectura de software.

El paradigma MVC, es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario, separando estas últimas de la lógica de negocio. La lógica de negocio refiere a la serie de procesos que se ejecutan cuando el usuario acciona la interfaz para realizar acciones

Capítulo 4. Diseño del sistema

con el programa. Estos procesos no son visibles por el usuario [12] [13].

El paradigma MVC surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, que potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos. En este paradigma se separa lo que refiere a la representación de los datos (Modelo), de la presentación de los mismos (Vista) y del flujo de operación y procesamiento que a estos se les realiza (Controlador).

El Modelo es la parte referida a la gestión del acceso de la información. Es quien envía a la vista aquella parte de la información que en cada momento se solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al modelo a través del controlador.

La Vista presenta en la interfaz la información brindada por el Modelo y es a través de esta interfaz con quien interactúa el usuario. Realiza una presentación de los datos del Modelo estando separada de los objetos del Modelo. Es responsable del uso de la información de la cual dispone para producir cualquier interfaz de presentación de toda petición que se presente [14]. Es lo que muestra y lo que recibe a través de clics en botones, menús, formularios, etc.

Por último el Controlador funciona como intermediario entre el Modelo y la Vista. Responde a eventos e invoca peticiones al Modelo cuando se hace alguna solicitud sobre la información. También puede enviar comandos a la Vista asociada si se solicita un cambio en la forma en que se presenta el Modelo.

La figura 4.1 presenta un esquema donde se muestra la interacción entre los componentes, las líneas punteadas indican una asociación indirecta y las enteras una directa.

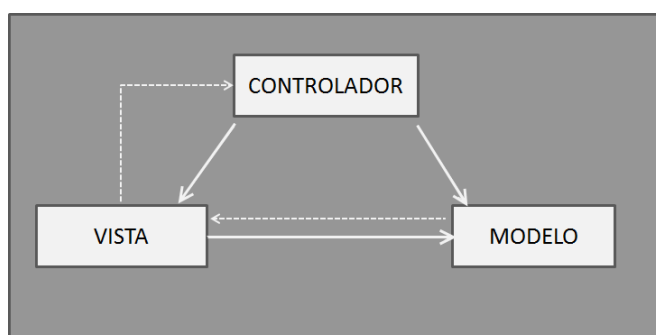


Figura 4.1: Interacción entre los componentes en el modelo MVC. Fig. extraída de [15]

El usuario interactúa con la aplicación a través de la interfaz de usuario (Vista), luego el Controlador, al recibir la notificación de dicho evento, decide qué hacer a través de su gestor de eventos (handler o callback). Eventualmente, dependiendo del tipo de evento, actualizará el Modelo. Por último el controlador delega a la

4.4. Modelo MVC de la aplicación KLEARapp.

vista la tarea de desplegar los datos en la interfaz de usuario, tomando del modelo los datos que se necesite.

Muchos sistemas informáticos utilizan un Sistema de Gestión de Base de Datos (SGBD) para gestionar los datos que debe utilizar la aplicación. Un SGBD es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. En el caso del modelo MVC dicha gestión corresponde al modelo.

4.4. Modelo MVC de la aplicación KLEARapp.

En el caso de KLEARapp, a partir del análisis de los requerimientos y de los diagramas de casos de uso, se armó el diagrama que corresponde al paradigma MVC. En la figura 4.2 se muestra un esquema con los tres componentes con sus archivos correspondientes. De esta forma se estructura la aplicación. Las líneas continuas indican las asociaciones directas entre las capas del modelo. Un controlador puede indicarle a su vista correspondiente que despliegue o solicite diversos tipos de datos.

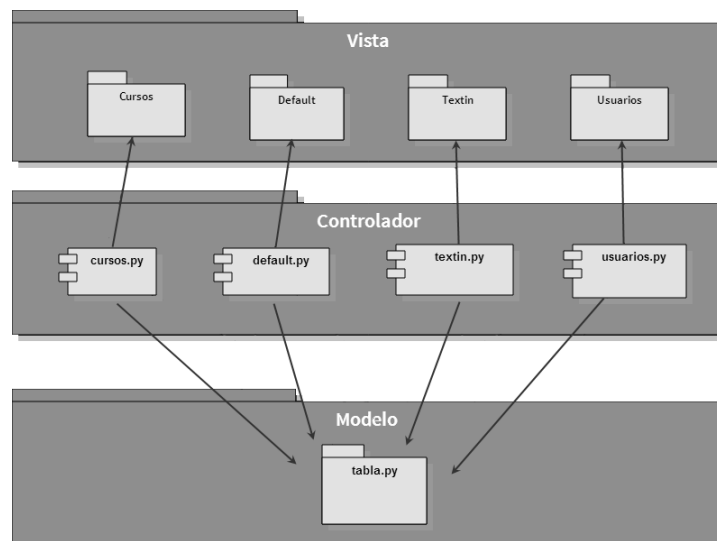


Figura 4.2: Diagrama de componentes del modelo MVC.

En la Vista se presentan los diferentes cuadros que debe mostrarse en la interfaz de usuario según cada evento. Luego estas se asocian con una función específica de un controlador que gestiona los datos de entrada y salida con dicha interfaz. Por último se exhibe el Modelo, donde se actualizan y modifican los datos.

Capítulo 4. Diseño del sistema

En las figuras 4.3, 4.4 y 4.5 se detallan las vistas asociadas a las funciones que llevan el mismo nombre (con extensión .py en lugar de .html) dentro del controlador en que son listadas.

Estas funciones se presentan en las figuras 4.6, 4.7 y 4.8 En la figura 4.9 se listan las tablas pertenecientes a las bases de datos `tablas.py`.



Figura 4.3: Vistas asociada a las funciones contenidas en el controlador `cursos.py`.

En el entendido que se llamará paquete al conjunto de las funciones contenidas en un controlador y a las vistas asociadas a las mismas, y módulo al par función-vista asociada, se describen a continuación las funcionalidades de cada paquete.

Los módulos del paquete `Cursos` permiten crear, modificar y eliminar cursos (módulos `formulario_curso`, `eliminar_curso`, `modificar_curso`, `elim_cursoadmin`), crear, modificar y eliminar unidades de aprendizaje (módulos `formulario_ua`, `modificar1_ua`, `modificar2_ua`, `eliminar_ua`), crear tareas (módulo `formulario_tarea`), crear el borrador donde el usuario puede probar escribir y generar grafos (módulo `formulario_borrador`), y generar el espacio para formular la evaluación (módulo `formulario_evaluacion`, `modificar_evaluacion`). Los módulos que permiten al administrador modificar los archivos de léxicos y reglas son `archivo_lex` y `archivo_rul`. El módulo que permite listar todos los cursos existentes es `index`.

Los módulos del paquete `Textin` permiten elegir el recurso al que se quiere

4.4. Modelo MVC de la aplicación KLEARapp.

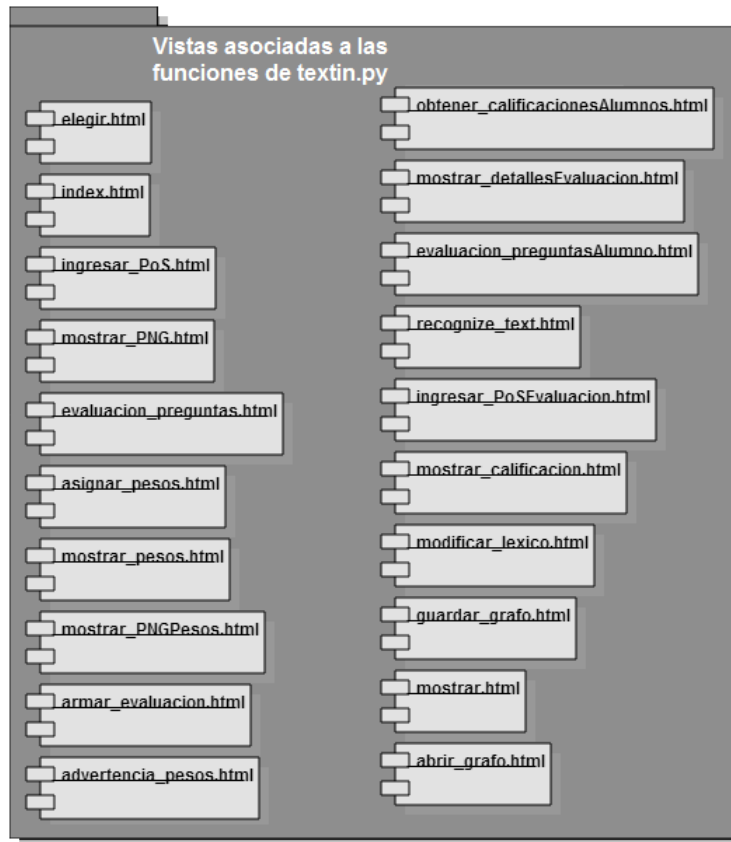


Figura 4.4: Vistas asociada a las funciones contenidas en el controlador textin.py.

acceder dentro de un curso (módulo elegir), realizar las acciones para la validación tanto del léxico (módulos `accept_text`, `modificar_lexico`, `ingresar_POS`, `ingresar_POSEvaluacion`) como de las reglas (módulos `index`, `recognize_text`, `recognize`), mostrar y guardar el grafo generado (`guardar_grafo`, `mostrar_PNG`) abrir y mostrar un grafo guardado (`abrir_grafo`, `mostrar`), asignar pesos a los nodos del texto (`asignar_pesos`, `mostrar_pesos`, `mostrar_PNGPesos`), generar las preguntas para la evaluación (luego de asignar los pesos) y el ingreso de las respuestas dadas por los alumnos (`advertencia_pesos`, `evaluacion_preguntas`, `armar_evaluacion`, `evaluacion_preguntasAlumno`) y mostrar las calificaciones obtenidas a docentes y alumnos (`obtener_calificacionesAlumnos`, `mostrar_detalleEvaluacion`, `mostrar_calificacion`).

El módulo del controlador `Default` contiene las funcionalidades para el registro e inicio de sesión en la aplicación.

En el controlador `Usuarios` se encuentran los módulos para matricularse, desmatricularse y reactivar a un usuario en los cursos (`matriculacion`, `desmatricularse`, `reactivar`), listar los participantes y las solicitudes para ingresar al curso (`alumnos`, `docente`, `solicitudes`). También presenta la pantalla para las funcionalidades del

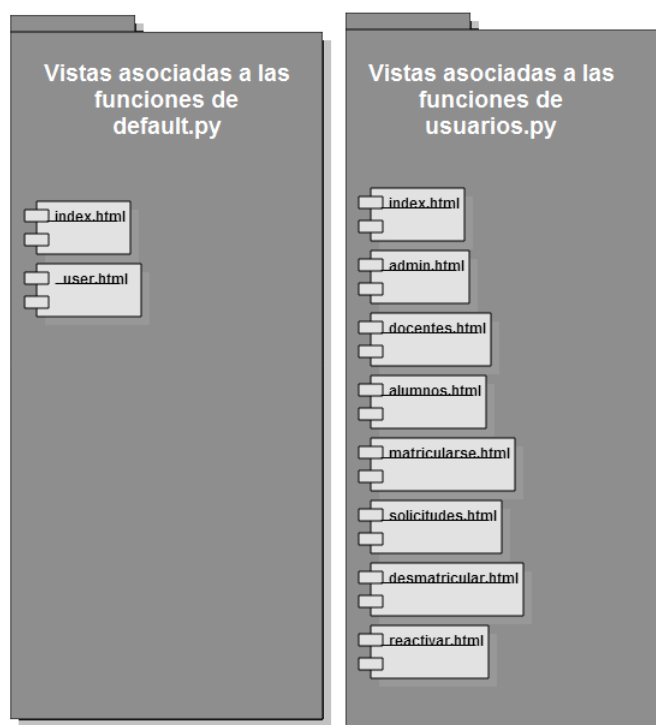


Figura 4.5: Vistas asociada a sus respectivos controladores.

administrador (admin).

A partir del Modelo MVC se desarrolló el código de la aplicación. En el capítulo 6 se detalla cómo se implementa el presente modelo en la aplicación KLEARapp y cómo se integraron los módulos del sistema KLEAR.

4.5. Componentes del sistema KLEARapp.

Para cumplir con el objetivo de dar soporte al sistema KLEAR, resulta valioso que el mismo esté organizado en una jerarquía de módulos escritos en Python. Web2py permite el uso de módulos externos escritos en este lenguaje, importándolos explícitamente.

Los mismos fueron ubicados en la carpeta MÓDULOS que viene provista en el framework como se representa en las figuras 4.10 y 4.11, de esta forma los módulos se copian y distribuyen de forma automática con la aplicación.

4.5. Componentes del sistema KLEARapp.

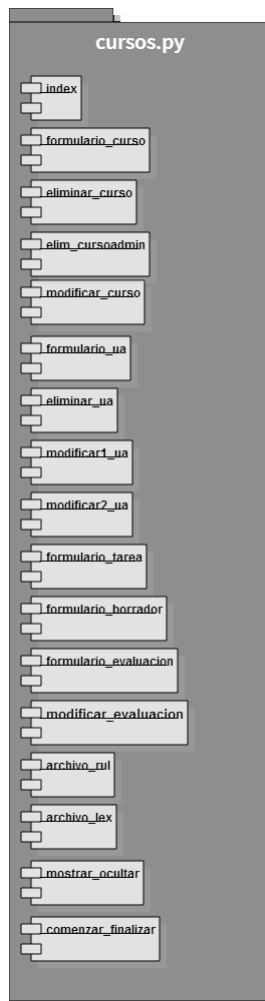


Figura 4.6: Funciones contenidas en el controlador `cursos.py`.

Una vez analizados los requerimientos del cliente se escribió los casos de uso en un formato estándar, que permite la transferencia de conocimientos a terceros. Luego se estudió y se determinó cuáles tecnologías se adaptaban mejor y se diseñó en base al paradigma MVC la solución propuesta.

De esta manera, se identificó la información perteneciente al Modelo - Vista- Controlador que interactúan directa e indirectamente entre sí. Se resolvió la integración del sistema KLEAR a través de la inclusión de sus módulos en el Modelo del framework.

Capítulo 4. Diseño del sistema

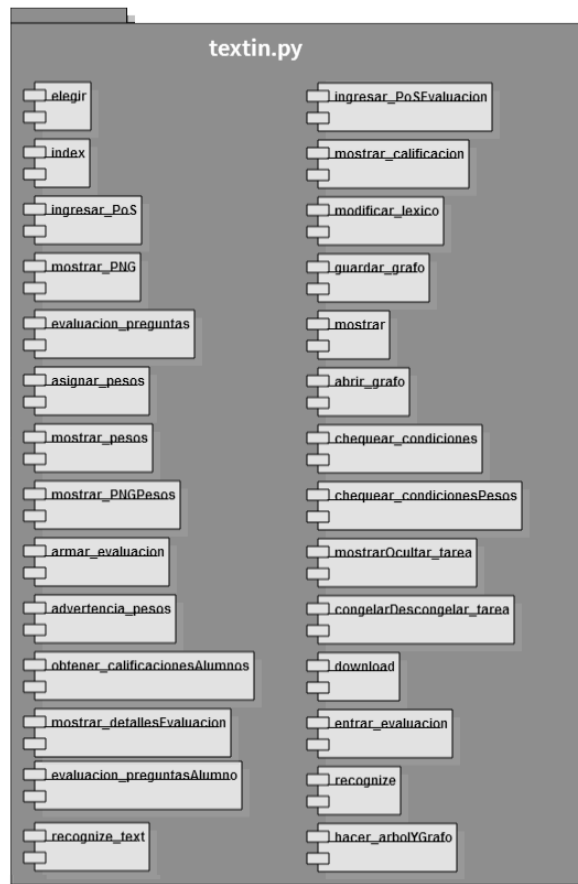


Figura 4.7: Funciones contenidas en el controlador textin.py.

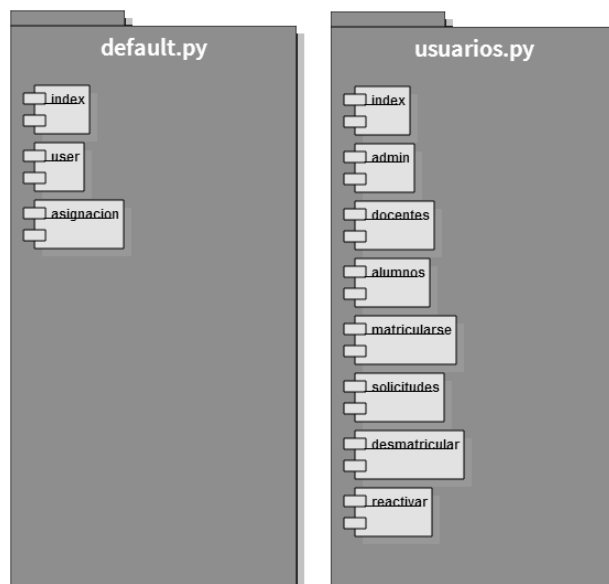


Figura 4.8: Funciones contenidas en los controladores default.py y usuarios.py.

4.5. Componentes del sistema KLEARapp.

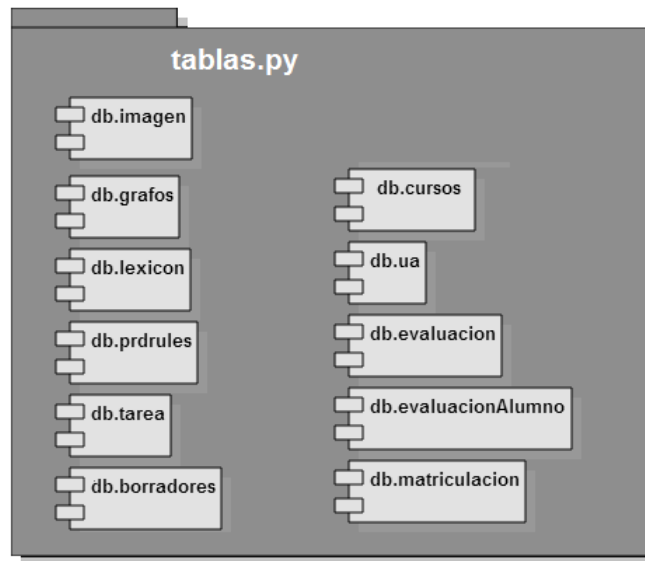


Figura 4.9: Tablas correspondientes a la base de datos propuesta en el Modelo.

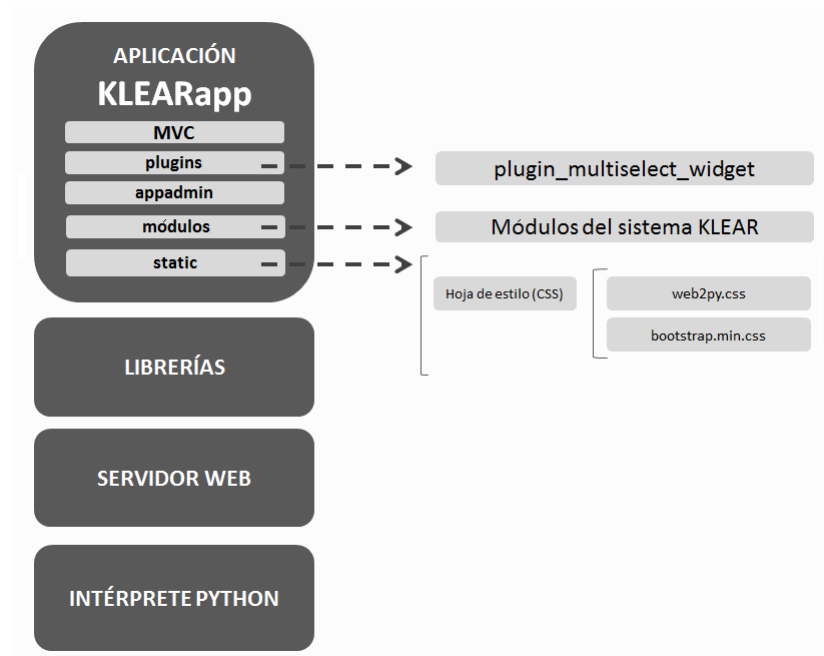


Figura 4.10: Componentes del sistema KLEARapp.

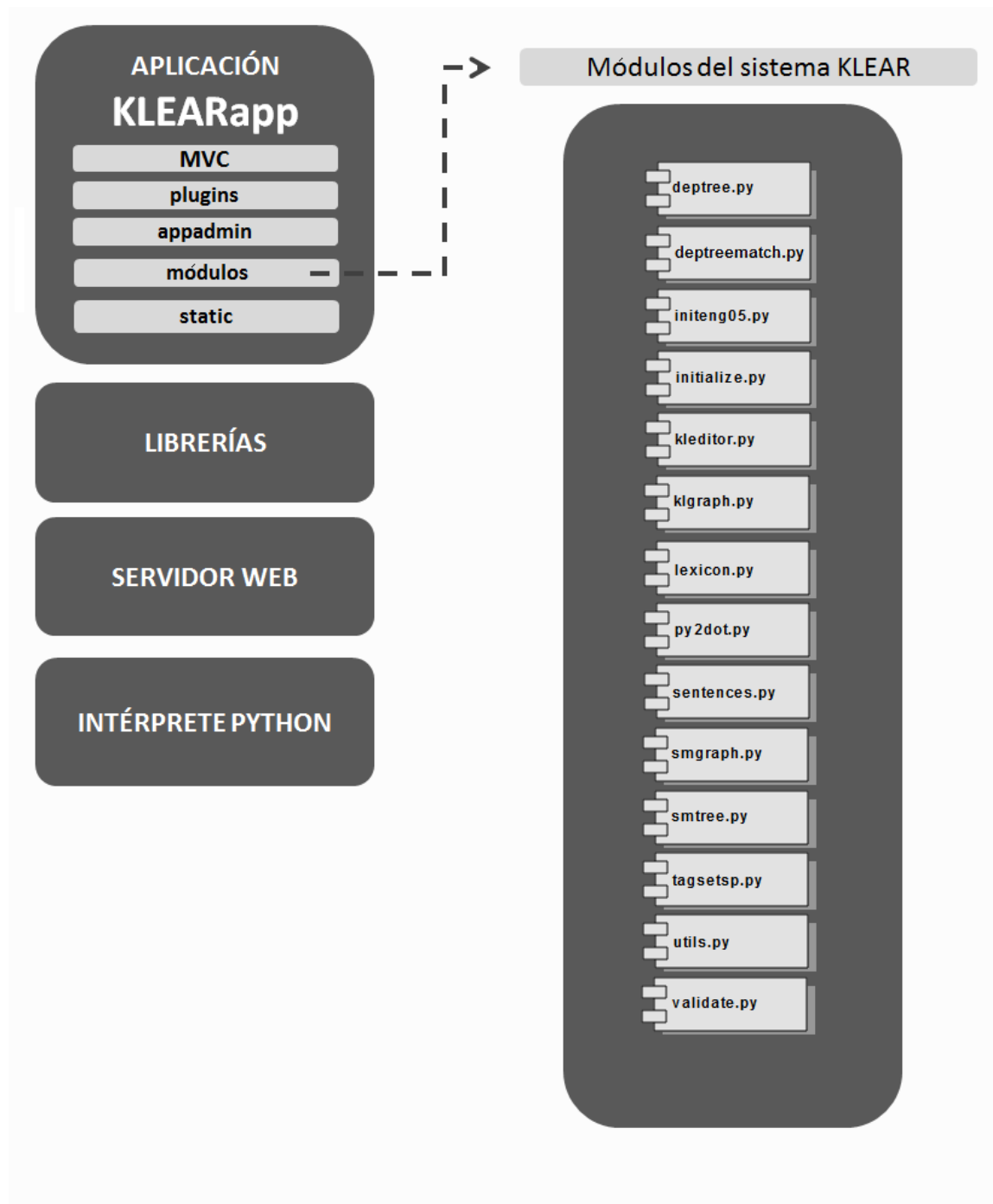


Figura 4.11: Detalle de los módulos del sistema KLEAR.

Capítulo 5

Modelo de base de datos.

5.1. Introducción.

En este capítulo se describe el significado de los modelos de bases de datos y cuál es el esquema del modelo relacional que se define para el diseño.

Luego se especifica qué motor de bases de datos se utiliza para la implementación.

5.2. Modelo de base de datos.

Los modelos de base de datos sirven como análisis abstracto de los datos, describiendo las relaciones y restricciones que deben cumplirse sobre los mismos.

Los modelos se pueden clasificar en distintos tipos dependiendo de su nivel de abstracción: los de niveles más altos, se llaman modelos conceptuales y los de bajo nivel son los modelos físicos. Luego también existe un nivel intermedio que se los llama modelos lógicos.

Se habla de alto nivel cuando se describe el comportamiento de los datos respecto a cómo lo ve el usuario.

En el proyecto se trabajó sobre el modelo conceptual. Para ello es importante identificar los siguientes conceptos: Entidades, Atributos y Relaciones.

- Entidad es cualquier objeto o concepto del mundo real.
- Atributo es una propiedad de alguna entidad.
- Relación describe la interacción entre dos o más entidades.

Capítulo 5. Modelo de base de datos.

Cuando se utiliza un modelo de base de datos para describir los comportamientos de los datos, se dice que se utiliza un esquema de la base de datos. Este esquema se debe especificar en la parte del diseño. En el proyecto, se realizó formalmente luego de terminado el mismo y queda como trabajo futuro su debida implementación.

Una vez configurado el esquema de la base de datos se prosigue a cargar los datos, los cuales pueden ir modificándose y actualizándose a través del tiempo. Por eso se habla del “estado de la base de dato” haciendo referencia a datos que se encuentran en el esquema en un determinado momento.

Luego el sistema de gestión de la base de datos (SGBD), que en este caso el mismo se encuentra incluido en el framework, es quien se encarga de garantizar que cada estado cumpla con las especificaciones y restricciones que se establecieron en el esquema [8] [16].

5.3. Modelo relacional.

Para realizar el esquema de las bases de datos, se utilizó un Modelo relacional. Éste es de los modelos conceptuales el más utilizado.

Para este modelo se resaltan siete conceptos fundamentales que permitirán la construcción del diseño del esquema de la base de datos.

Se define la “relación” a través de una tabla de filas y columnas. Cada columna se corresponde con un atributo y cada fila con lo que se conoce como registro individual. Los atributos tienen valores que se encuentran definidos en un cierto Dominio.

Cuando se habla de una fila en particular, se está hablando de una “tupla” de la tabla. Luego la cantidad de atributos que tiene una tabla se la llama “grado de una relación” y el número de tuplas, “cardinalidad de una relación”.

Para especificar el esquema del modelo relacional es importante introducir el concepto de claves.

Se define superclave al atributo o conjunto de éstos que permite identificar de forma única a las tuplas dentro de una relación. Una clave candidata son aquellas superclaves en donde cualquier subconjunto de estas no asegura la unicidad de las tuplas.

Luego se define clave primaria a aquella dentro del conjunto de claves candidatas que se elige para identificar de forma única a las tuplas de la relación. Y las restantes claves candidatas se definen como alternativas.

5.4. Esquema de una base de datos relacional.

Por último se encuentra la clave ajena que es un atributo o conjunto de atributos de una relación cuyos valores coinciden con los valores de la clave primaria en alguna otra relación [8] [16].

5.4. Esquema de una base de datos relacional.

A partir de los conceptos definidos en las secciones anteriores y de los datos extraídos de lo realizado en el desarrollo, se listan los nombres de las relaciones con las que se trabaja en la aplicación y se especifican los atributos. A su vez para cada relación, se subraya aquel atributo que se identificó como clave primaria.

USUARIO (nombre, apellido, mail_usuario, cédula, contraseña)

CURSOS (id_curso, nombre, descripción)

MATRICULACIÓN (mail_usuario, cédula, nombre_completo, curso, rol, estado, inicio, fin)

TAREA (nombre, descripción, ua_nid, texto_referencia, estado_pesos, klgrRef, mostrar, descongelar)

BORRADORES (nombre, ua_nid, texto, klgrBorrador, cédula)

UA (título, descripción, ua_nid, id_curso)

EVALUACIÓN (nombre, ua_nid, preguntas, mostrar, respuestas, comenzar)

EVALUACIÓN_ALUMNO (id_curso, ua_nid, cédula, nota, respuesta)

Como se puede observar, las relaciones TAREA, EVALUACIÓN y UA tiene la misma clave primaria: ua_nid. Se podría pensar en unir dichas relaciones en una, llamándola de la siguiente forma:

UA (título, descripción, ua_nid, id_curso, tarea_nombre, tarea_descripción, texto_referencia, tarea_estado_pesos, tarea_klgrRef, tarea_mostrar, tarea_descongelar, evaluación_nombre, evaluación_preguntas, evaluación_mostrar, evaluación_respuestas, evaluación_comenzar)

Con el mismo razonamiento podríamos agrupar las relaciones BORRADORES Y EVALUACIÓN_ALUMNO:

ACTIVIDAD_ALUMNO (ua_nid, cédula, id_curso, borrador_nombre, borrador_texto, borrador_klgrBorrador, evaluación_alumno_nota, evaluación_alumno_respuesta).

Capítulo 5. Modelo de base de datos.

En resumen quedaría el siguiente esquema:

USUARIO (nombre, apellido, mail_usuario, cédula, contraseña)

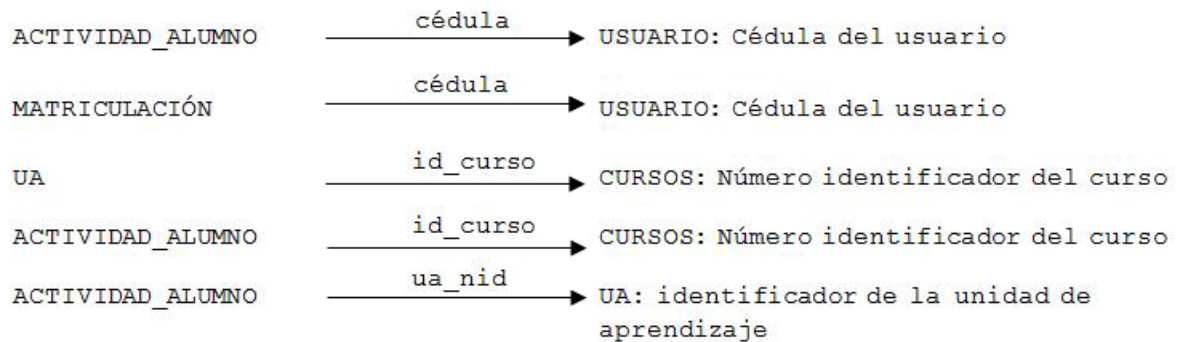
CURSOS (id_curso, nombre, descripción)

MATRICULACIÓN (mail_usuario, cédula, nombre_completo, curso, rol, estado, inicio, fin)

UA (título, descripción, ua_nid, id_curso, tarea_nombre, tarea_descripción, texto_referencia, tarea_estado_pesos, tarea_klgrRef, tarea_mostrar, tarea_descongelar, evaluación_nombre, evaluación_preguntas, evaluación_mostrar, evaluación_respuestas, evaluación_comenzar)

ACTIVIDAD_ALUMNO (ua_nid, cédula, id_curso, borrador_nombre, borrador_texto, borrador_klgrBorrador, evaluación_alumno_nota, evaluación_alumno_respuesta).

A continuación se presentan los diagramas referenciales. El atributo encima de la flecha es clave ajena de la relación origen y clave primaria de la relación destino.



Capítulo 6

Implementación

6.1. Introducción.

En este capítulo se describe la codificación implementada para resolver aspectos significativos de la aplicación. Se explica como se implementan las funcionalidades que debe cumplir el sistema.

6.2. Autenticación.

La autenticación define qué pantallas se muestran al usuario dependiendo de su rol. La misma involucra el control de acceso a la aplicación. Este control se realiza en base al mecanismo que utiliza web2py de “control de acceso basado en roles” (RBAC por sus siglas en inglés) que tiene como finalidad restringir el acceso al sistema únicamente a usuarios autorizados.

Haciendo uso del RBAC se otorgan ciertos permisos a determinados grupos de usuarios dependiendo de las funcionalidades que deban ejecutar, es decir que se tienen distintos controles sobre las actividades dependiendo del rol (Alumno, Docente y Administrador). En web2py se puede definir primero los grupos haciendo coincidir cada uno con un rol diferente y luego decidir a cuál pertenece cada usuario. En las librerías de web2py se definen funciones con este objetivo. La clase `Auth` incluye un conjunto de métodos que interactúa con tablas específicas para este propósito, estas son: `db.auth_user`, `db.auth_membership` y `db.auth_group`.

En la tabla `db.auth_group` se definen los grupos. Por defecto el sistema web2py crea un grupo por usuario. Dado el propósito de la aplicación se modifica su configuración para definir los grupos como se muestra en la figura 6.1.

En la tabla `db.auth_user` se guardan todos los datos de aquellos usuarios que se registran en la aplicación. El administrador tiene la particularidad de tener un valor predeterminado para el campo “First name”- Admin.klear2014-. Con este

Capítulo 6. Implementación

```
if not db(db.auth_group).count():
    db.auth_group.insert(role='docente',description='Administra cursos')
    db.auth_group.insert(role='alumno',description='Participa de los cursos')
    db.auth_group.insert(role='admin',description='Administra docentes')
```

Figura 6.1: Código que define los grupos en el modelo tablas.py.

dato queda registrado en `db.auth_membership` con rol administrador.

La tabla `db.auth_membership`, es la que asocia un usuario específico a través de un identificador de `db.auth_user` con uno de los grupos creados.

Para alumnos y docentes, el registro en `db.auth_membership` se realiza en función de una variable que toma valor “Alumno” o “Docente” dependiendo del botón seleccionado en la pantalla principal de la aplicación como se muestra en la figura 6.2.

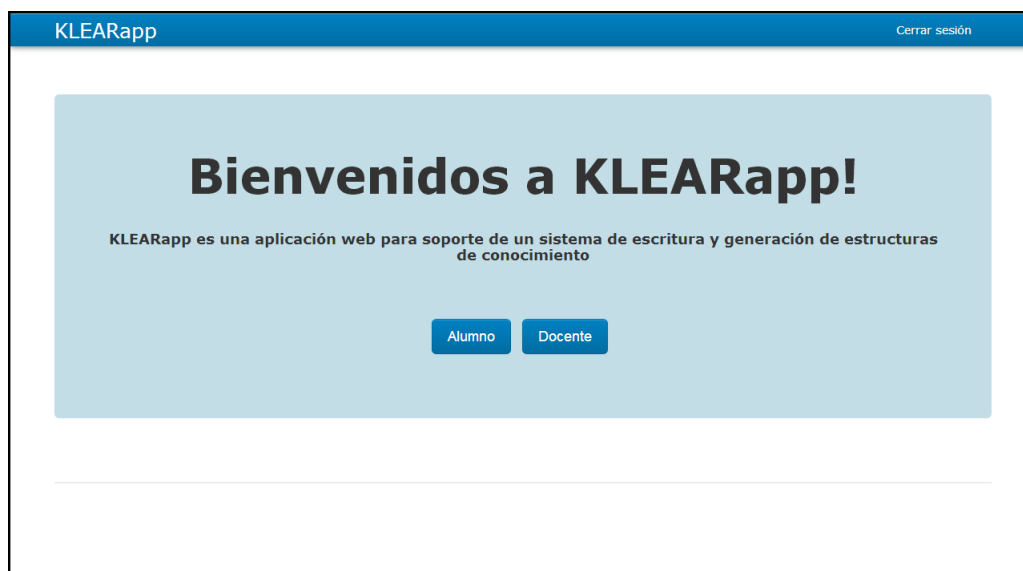


Figura 6.2: Vista de la pantalla principal de la aplicación.

6.3. Matriculación.

La matriculación es el medio por el cual los alumnos y docentes son considerados como posibles participantes de un curso.

Con este fin se define una tabla en la base de datos, `db.matriculacion` cuyos campos se formaron a partir de los datos del usuario, los datos del curso involucrado y el estado que establece la relación entre el usuario y el curso. Estos estados

6.4. Integración del sistema KLEAR a la aplicación KLEARapp.

son activo, borrado y pendiente.

En estado activo un usuario tiene permiso para acceder a los recursos del curso. En estado pendiente se encuentra quien ha solicitado permiso para matricularse y está a la espera de ser aceptado ó no. En estado borrado el usuario no puede tener acceso al curso.

Dependiendo del estado se determinan las distintas acciones que se describieron en los casos de usos relacionados con la matriculación. Para implementar lo anterior se creó una función con sentencias condicionales que evalúan este campo y conforme a su valor, se despliegan los mensajes alusivos a la situación del usuario frente al curso y se redirige a la pantalla correspondiente. El sistema ejecuta esta función cada vez que el usuario intenta ingresar a un curso.

6.4. Integración del sistema KLEAR a la aplicación KLEARapp.

La incorporación del sistema KLEAR en la aplicación KLEARapp se logra agregando sus módulos en la carpeta “Modelos” del framework, e invocándolos con el comando `import` desde el controlador `textin.py`. Esto permite llamar a las funciones del sistema KLEAR, utilizando como entrada los datos obtenidos de la interfaz de usuario.

6.5. Ingreso de texto y generación de grafo.

Para generar el grafo de un texto en particular, primero se crean formularios como medio para que el alumno o docente interactúe con el sistema a través de la interfaz de usuario. La información obtenida se utiliza como entrada de las funciones que componen al sistema KLEARapp y que invocan las correspondientes al sistema KLEAR. De esta forma se valida el texto: en primera instancia se realiza la validación de léxico, para asegurar que toda palabra tenga una única categoría léxica y luego las reglas que permiten asegurar la correcta semántica. Concluida esta etapa se genera la estructura abstracta que contiene toda la información para crear el grafo.

Tanto el texto que ingresa el usuario como la estructura abstracta - objeto `klgraph`- se guardan en las tablas de la base de datos asociadas a la actividad correspondiente: `db.tarea`, `db.borradores` y `db.evaluacion`.

El uso de tablas permite que la información se guarde independientemente de la sesión de usuario y recuperarla en cualquier momento. Esto es útil para crear

Capítulo 6. Implementación

un texto en forma colaborativa por todos los participantes, donde cada uno edita y el sistema actualiza esta información en la tabla correspondiente y se recupera para una posterior edición.

El texto por ser de un tipo de objeto conocido, “text”, se guarda como string. En cambio la estructura abstracta es una instancia de un objeto de clase definida por el sistema KLEAR, de tipo desconocido para web2py. Por eso se utiliza uno de los módulos que trae incorporado Python que se llama pickle, que convierte un objeto a un formato conocido que se pueda almacenar y recuperar desde la tabla. La figura 6.3 muestra un fragmento del código que muestra cómo serializar un objeto klgraph.

```
f=db(db.grafos.id==request.args[0]).select()[0]
fi=os.path.join(request.folder, 'uploads', f.ob_kgr)

fin=open(fi,'rb')
klgr=pickle.load(fin)
fin.close()

fout=open('graph0b', 'wb')
pickle.dump(klgr, fout)
fout.close()
```

Figura 6.3: Fragmento del código que muestra cómo serializar un objeto klgraph.

6.6. Asignación de pesos

El sistema KLEAR tiene definido pesos iniciales para cada categoría de léxico. La aplicación KLEARapp incluye una función que permite al docente modificar o agregar nuevos pesos a los nodos. Los arcos no son ponderados porque en este tipo de estructuras su contenido semántico es muy bajo en relación a los nodos.

Se desarrolló un buscador que recorre dos listas, la de los nodos generados a partir del texto ingresado y la que corresponde a sus pesos, en una relación uno a uno en sus índices. El buscador presenta el nodo deseado con su peso y permite que el docente lo modifique a través de un formulario. De esta forma la aplicación permite asignar y modificar el peso a cada nodo del texto ingresado.

6.7. Mostrar - ocultar y congelar - descongelar

Para ocultar o mostrar alguna tarea o evaluación se utiliza un campo en las tablas que definen su estado (oculto o visible). Luego en el controlador textin.py se definió una función de sentencias condicionales, que comparan el campo del estado con una bandera. Dependiendo de este resultado y del rol del usuario se visualizan

6.8. Construcción de la evaluación.

u ocultan las actividades y se presentan de diferentes formas.

Cuando se trata de un docente, el sistema muestra la tarea con un ícono adjunto que indica su estado como se muestra en las figuras 6.4 y 6.5. En el caso de un alumno, dependiendo del estado se mostrará o no dicha actividad.

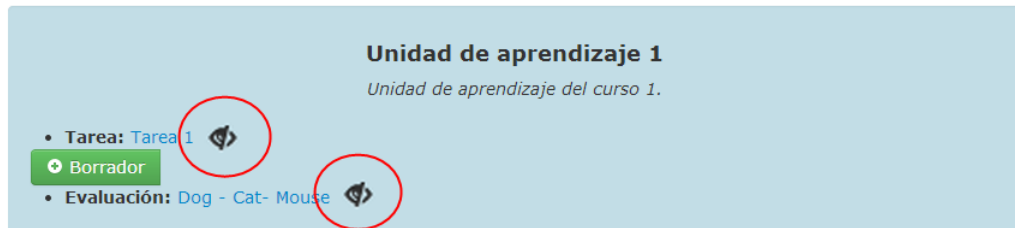


Figura 6.4: Tarea oculta por el docente. El ícono muestra un ojo cerrado destacado con un círculo.

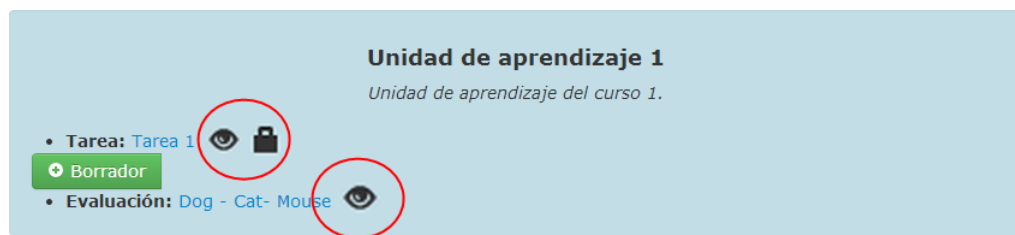


Figura 6.5: Tarea visible y congelada. Se destacan los íconos ojo abierto y candado.

Para descongelar o congelar una tarea se procedió de la misma forma que mostrar o ocultar, y además en el caso del alumno se deshabilita el área de edición. Esto se realiza mediante la siguiente sentencia que se muestra en la figura 6.6.

```
if auth.has_membership('alumno') &
    ( db(db.tarea.ua_nid==uanid).select()[0]['descongelar']==None):

    tarea= db(db.tarea.ua_nid==uanid).select()[0]
    db.tarea.textoref.writable=False
```

Figura 6.6: Fragmento del código que permite que un campo de un formulario no sea editable.

6.8. Construcción de la evaluación.

La implementación de la actividad evaluación fue la más compleja de realizar en esta aplicación, porque en su lógica se integran códigos de los recursos vistos anteriormente. La evaluación como actividad está particionada en lo que hace el

Capítulo 6. Implementación

docente para formularla y por otro lado cómo accede el alumno para contestar esta evaluación.

Se crean dos tablas en la base de datos. Una que concierne a la elaboración de la evaluación por parte del docente, `db.evaluacion` y otra que incluye las respuestas de cada alumno, `db.evaluacionAlumno`. Ambas tablas tienen campos que identifican el nombre de la evaluación, a qué curso y unidad de aprendizaje corresponde.

La `db.evaluacion` además contiene un campo “lista_preguntas”, de tipo `list`, donde se almacenan todas las preguntas que se quiere formular. También posee un campo `file` de nombre “respuesta”, formado por un diccionario que es un tipo de colección de datos, definido en el lenguaje Python, que relaciona cada ítem de la colección con una clave y un valor. En este caso la clave es alguna de las preguntas y el valor está formado por una tupla (tipo de colección muy parecido a una lista) que se forma con el texto de la respuesta y el objeto `klgraph` asociado al texto.

El docente crea la evaluación y registra las preguntas a través de un formulario, al enviar los datos se almacenan en la tabla. Luego arma la respuesta a partir del texto referencia almacenado en la tabla `db.tarea`. Para ello se listan las sentencias y se ingresan como entrada en un selector, implementado en el plugin `plugin_multiselect_widget` para elegir aquellas que conforman la respuesta.

Los plugin son archivos que pueden contener un modelo, controlador o vista que se instala encima de la aplicación [8]. El `plugin_multiselect_widget` fue extraído de la página Scubism <http://dev.s-cubism.com/>. Una vez descargado en la carpeta “Plugin” es importado en el controlador `textin.py`. En la figura 6.7 se muestra cómo se importa la función desde el plugin al controlador `textin.py`. Y en la figura 6.8 se muestra el fragmento de código donde se ejecuta esta función.

```
from plugin_multiselect_widget import (vmultiselect_widge)
```

Figura 6.7: Fragmento del código donde se importa el plugin en el controlador `textin.py`.

```
db2.Enunciados.Enunciados.widget = vmultiselect_widget
```

Figura 6.8: Fragmento del código que muestra cómo se invoca a la función del plugin importado.

En la figura 6.9 se muestra la pantalla con el `plugin_multiselect_widget` integrado.

Una vez que la evaluación se encuentra visible para el alumno, éste ingresa y contesta las preguntas. El sistema almacena en uno de los campos de la tabla `db.evaluacionAlumno` un diccionario donde la clave es la pregunta y el valor es una tupla que incluye el texto ingresado como respuesta por el alumno, su objeto

The screenshot shows the KLEARapp interface. At the top, there is a blue header with 'KLEARapp' on the left and 'Bienvenido Docente' on the right. Below the header, there are three navigation menus on the left: 'Menú Participantes' with options 'Lista alumnos', 'Lista docentes', and 'Desmatricularme'; 'Menú Administrativo' with options 'Administrar curso', 'Solicitudes', and 'Reactivación'; and 'Menú General' with options 'Léxico del curso', 'Curso actual', 'Cursos', and 'Ayuda'. The main content area is titled 'curso1' and 'Evaluación: Dog - Cat- Mouse'. Below this, it says 'Pregunta 1: Whose is the cat?'. There is a text area containing the question text with brackets around words: 'the big black dog chased [the thin yellow cat] [to the house [in the street]] [yesterday [in the morning]] the food lay [in the house] the mouse ate [the food] the cat killed [the mouse] the cat belonged [to the man [with the guitar]]'. Below the text area are two buttons: 'Agregar' and 'Quitar'. At the bottom of the main area, there are three buttons: 'Guardar respuesta', 'Volver a la lista de preguntas', and 'Volver al curso'.

Figura 6.9: Pantalla donde se muestra el plugin con las sentencias para armar la respuesta.

klgraph, el peso de los nodos reconocidos en el texto respuesta dada por el docente y el total de los pesos asignados a los nodos de este último.

6.9. Testing.

El testing es una actividad desarrollada para evaluar la calidad del producto, y para mejorarlo al identificar defectos y problemas. El testing de software consiste en la verificación dinámica del comportamiento de un programa sobre un conjunto finito de casos de prueba, apropiadamente seleccionados a partir del dominio de ejecución que usualmente es infinito, en relación con el comportamiento esperado [17].

La verificación es dinámica debido a que el programa es testeado poniéndolo en ejecución de la forma más parecida posible a como se utilizará cuando esté en producción.

Si uno de los casos de prueba detecta un error: el programa es incorrecto; pero, si en ninguno de los casos de prueba seleccionados encuentra un error: no podemos decir que el programa es correcto en el sentido de perfecto pero sí correcto en el marco de los casos de prueba seleccionados [17].

6.9.1. Unit testing.

Como se observa en la figura 6.10, el software puede testearse a diversos niveles. La metodología de realizar pruebas de unidad (a continuación se definirá el concepto de unidad) conocida por su nombre en inglés como Unit Testing, es la primera en emplearse una vez que comienza la etapa de desarrollo.

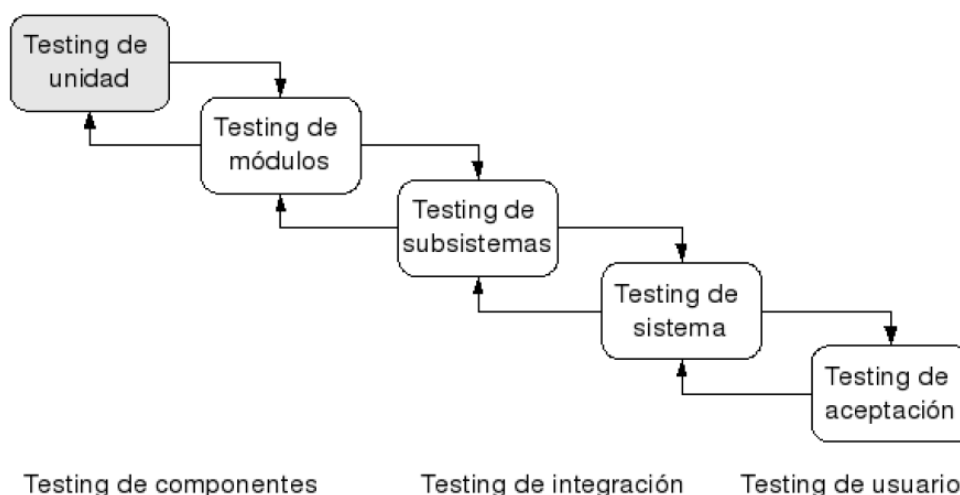


Figura 6.10: El proceso general de testing. Figura extraída de [17].

Unit testing es un método de testing de software que tiene como objetivo comprobar el correcto funcionamiento de unidades individuales de código, entendiendo por unidad una porción de código que puede aislarse del resto y probarse su funcionalidad [18]. En programación orientada a objetos, una unidad es por lo general una entidad entera, como una clase, pero puede también tomarse como un método [19].

Este procedimiento permite chequear que las unidades funcionen correctamente por separado. Luego, las Pruebas de Integración serán las encargadas de garantizar el correcto funcionamiento de un conjunto de éstas que interactúen entre sí [20].

Las características que se le suelen imponer a estas pruebas de unidad son [20]:

- **Automatizables:** Deben de poder ejecutarse sin intervención manual.
- **Completas:** Deben cubrir la mayor parte de código.
- **Reutilizables:** Deben de poder ejecutarse cuantas veces se desee.
- **Independientes:** Deben de poder ejecutarse de forma individual, en cualquier orden, sin dependencias entre ellas.
- **Profesionales:** Deben de ser documentadas y tratadas con igual profesionalismo que el código.

Como se explicó en el capítulo 4, KLEARapp está diseñada siguiendo el paradigma de arquitectura MVC. Así, cuenta con los componentes del Modelo, los de la Vista y los del Controlador. Como se explicó en dicho capítulo, estas tres partes interactúan entre sí de forma casi continua: comenzando por la Vista que recibe estímulos del usuario, siguiendo por el Controlador que decide qué hacer con éstos estímulos, pudiendo eventualmente realizar consultas o modificaciones en el Modelo, y da información a la Vista para que ésta se encargue de actualizar la interfaz de usuario acorde a los estímulos recibidos, comenzando nuevamente así el ciclo de descripto.

De lo antes dicho se puede inferir que los sistemas que emplean el modelo MVC basan su funcionamiento en la interacción hombre-máquina y, por ello, para testearlos no se puede emplear Unit Testing dado que las pruebas no cumplen con la característica de automatismo. Este método sí es válido para probar los módulos cargados en la aplicación, ya que los mismos son archivos de Python que definen funciones y/o clases con sus respectivos métodos, que pueden ser tratados como unidades y testearse su funcionalidad de forma automática sin que sea necesaria la intervención de personas durante la prueba.

Para la verificación del correcto funcionamiento de todo el sistema KLEARapp puede elaborarse un protocolo de testing que incluya los pasos a seguir para probar la funcionalidad de la aplicación en determinadas situaciones.

6.9.2. Protocolo de testing.

Este protocolo permite validar el comportamiento de KLEARapp frente a los distintos casos de uso. A continuación se detalla el procedimiento a seguir para testear la respuesta de la aplicación en un caso de uso.

La persona encargada de realizar el testing (tester) interpretará el papel de usuario final, realizando sobre la Vista las acciones que cada caso estipule.

Una vez que la prueba de todos los casos presentados en el Apéndice B finalice, hacer nuevamente una pasada por todos para chequear que los efectos de los cambios realizados durante los tests hayan sido correctamente estimados y, por lo tanto, se haya corregido de forma completa todos los casos afectados por esos cambios. Una vez hecho esto, y por lo tanto chequeado el funcionamiento global de KLEARapp, cada vez que se decida hacer una modificación, se debe hacer el testing sobre todos los caso de uso.

Testing de un caso de uso.

Los pasos a seguir para testear el comportamiento de KLEARapp en la situación de un caso de uso son:

Capítulo 6. Implementación

1. Seleccionar el caso de uso a testear
2. Chequear las precondiciones y verificar que cuando se intenta llevar a cabo el caso de uso:
 - 2.1. Si las precondiciones no se cumplen: la aplicación responda de forma apropiada; por ejemplo: dando mensajes que le avisen al usuario que no se puede realizar lo solicitado. En caso de no haber contemplado esta situación, implementar una solución y volver al paso 2. Además, testear después todos aquellos casos que se hayan visto afectados por la modificación del código.
 - 2.2. Si las precondiciones se cumplen: pueda comenzarse a probar lo establecido al comienzo del flujo básico del caso. Si esto no se puede realizar, implementar una solución y volver al paso 2. Además, testear después todos aquellos casos que se hayan visto afectados por la modificación del código.
3. Actuar de acorde a lo establecido en el flujo básico para el usuario, observando que la respuesta del sistema sea la estipulada en este flujo. En caso de encontrar un comportamiento fuera de lo documentado:
 - 3.1. Si lo implementado se considera más satisfactorio (en términos de funcionalidad y “amabilidad” con el usuario) que lo establecido en el caso: actualizar el caso en cuestión y todos los que correspondan de modo que reflejen el nuevo comportamiento. Además, aplicar luego el protocolo de caso en todos los casos afectados por el cambio.
 - 3.2. Si el resultado es un error reportado en la vista o no es satisfactorio (en los mismos sentidos antes mencionados), implementar una solución y volver al paso 2. También, aplicar después el protocolo de caso en todos los casos afectados por la corrección.
4. Una vez finalizada la validación del comportamiento de KLEARapp en el flujo básico del caso, probar los diferentes flujos alternativos comparando la respuesta obtenida con la documentada. En caso de obtener diferencias, proceder, según corresponda, de la misma forma que detallan los puntos 3.1. y 3.2.
5. Habiendo verificado el correcto funcionamiento del producto en los flujos alternativos, aplicar el procedimiento descrito en los pasos 1. a 5. a aquellos casos de uso que puedan haberse visto afectados por las eventuales modificaciones realizadas en 2. , 3. y 4.

Durante este proceso, a raíz de otros cambios que puedan surgir, habrá que chequear nuevos casos de uso. Esto puede llevar a pensar que este protocolo divergirá en duración y será así impracticable. Sin embargo, la forma en la que el paradigma MVC organiza el código, hace que las repercusiones que

6.9. Testing.

tengan las correcciones hechas a raíz de un caso de uso, sean las mínimas posibles sobre los restantes (evidentemente éstas estarán ligadas al grado interacción entre los casos). La estructura del modelo MVC también lleva a que en la etapa de desarrollo, a medida que se van terminando de implementar funcionalidades, se vayan haciendo pequeños tests desde el punto de vista del usuario (sumado, evidentemente, a los chequeos que pueden realizarse mediante la función “print” de Python que imprime en la consola de web2py lo que se le pase como argumento) que permiten ir verificando el comportamiento del sistema.

Capítulo 7

Despliegue

7.1. Introducción.

En este capítulo se detallan los programas requeridos para la instalación de la aplicación KLEARapp, y se enumera el paso a paso para la ejecución de los mismos.

7.2. Programas utilizados.

Como se ha mencionado anteriormente, la aplicación KLEARapp ha sido desarrollada web2py. Este entorno de desarrollo está escrito y es programable en el lenguaje de programación Python.

Python no está integrado en la carpeta de las distribuciones de código fuente de web2py, debe instalarse de forma separada para que el entorno de trabajo funcione.

A continuación se detallan los pasos que fueron seguidos para estar en condiciones de comenzar a desarrollar la aplicación. A saber, estas condiciones son: contar con el framework funcionando de forma correcta y tener exitosamente integrados las librerías y módulos de terceros, así como las herramientas necesarias para lograr los distintos objetivos de la aplicación.

Las librerías requeridas son las de NLTK (Natural Language Toolkit) [6]. Éstas permiten que los módulos del proyecto KLEAR operen correctamente a la hora de elaborar las distintas estructuras de representación del conocimiento.

Por otra parte, también se necesita adquirir un software que permita la generación de imágenes en formato PNG a partir de un archivo “dot text” (texto de puntos) en formato gv. Con este propósito, se emplea el ejecutable “dot.exe” que forma parte del software de visualización gráfico de código abierto Graphviz [21].

Capítulo 7. Despliegue

Vale aclarar que el procedimiento que sigue fue realizado en computadoras con sistema operativo (de ahora en más SO) Windows de 32 bits.

7.3. Detalle del paso a paso.

Pasos seguidos:

1. Se descargó e instaló la última versión de Python en la familia 2.x.y. En su momento, ésta fue la 2.7.6.
2. Se descargó y descomprimió alguna de las distribuciones de la última versión de web2py que fuera adecuada al SO. En <http://www.web2py.com/init/default/download> se puede consultar la disponibilidad del framework en las distintas plataformas de trabajo. En aquel entonces, la versión obtenida fue la 2.5.1.
3. Se instalaron las librerías del NLTK en una edición compatible con el Python instalado y el SO usado. Instructivo de instalación disponible en: <http://nltk.readthedocs.org/en/latest/install.html>. La misma se ubica en la carpeta site-packages de web2py.
4. Se colocó la carpeta descargada de Graphviz, graphviz-2.38 en: http://www.graphviz.org/Download_windows.php, también en el directorio del site-packages de web2py.
5. En el módulo “py2dot” (extraído del sistema KLEAR) de la aplicación en construcción, se cambió el parámetro de entrada “engine” por defecto y la variable “path” del método mkimgfile “(dotfile, outfile, engine, imgfmt)” de la siguiente manera:
 - engine=“dot.exe”
 - path=[Ruta a la carpeta que contiene el “dot.exe”], por ejemplo:
path=‘C: \KlearApp \Desktop \web2py \site-packages \release \bin’

Luego de estos pasos se pudo utilizar los módulos del sistema KLEAR sin inconvenientes.

Para poder ejecutar KLEARapp debidamente, habría que seguir las etapas previamente descriptas y agregar un paso 6 que constituya la instalación del paquete de la aplicación.

Se probó este procedimiento en un PC carente de contenidos previos que puedan favorecer el funcionamiento del instructivo detallado y se ha encontrado que: actualmente la última versión del framework es la 2.9.11. Ésto dificulta la integración del producto ya que aparecen una serie de tickets que la versión 2.5.1, con la

7.3. Detalle del paso a paso.

misma aplicación, no emite.

Dado el supuesto realizado al momento de planificar el proyecto, mencionado en el tercer punto de 6.2.7., se considera que la mejor manera de garantizar la correcta distribución del sistema es: proporcionar la versión del marco en la que la aplicación fue implementada. Así, se distribuye una carpeta de web2py con: la aplicación KLEARapp instalada, las librerías del NLTK integradas y comunicándose correctamente con el sistema, y el programa generador de imágenes PNG (“dot.exe” del Graphviz) incorporado.

Aquel que desee instalar KLEARapp, sólo deberá contar con alguna versión de Python de la familia 2.6.x - 2.7.x (para asegurar la compatibilidad con el NLTK disponible en la carpeta proporcionada) y cambiar en el módulo “py2dot” del sistema el “path” especificado en el punto 5.

Lo antes descrito fue exitosamente probado en un equipo con Windows 8 de 64 bits (con las dificultades que usar Python para 64 bits implica) con la versión apropiada de Python 2.7.8.

Se recomienda fuertemente emplear esta última opción debido a los inconvenientes mencionados y a otros que puedan surgir producto de las actualizaciones de las diversas herramientas respecto a las versiones empleadas en la elaboración de la aplicación.

En caso de presentarse algún inconveniente que impida la correcta ejecución del sistema, contactarse con el equipo de proyecto a soporteKLEARapp@gmail.com.

Capítulo 8

Evaluación

8.1. Introducción.

En este capítulo se realiza una evaluación del proyecto analizando en qué medida el Análisis de los casos refleja al producto final y verificar el cumplimiento de los criterios de éxito planteados al inicio de este proyecto.

8.2. Evaluación a partir del Análisis de los casos de uso.

Como se explicó en los capítulos anteriores el Análisis de los casos de uso se desprende de entender los requerimientos por parte del cliente, por lo que se puede considerar un buen indicador para evaluar la aplicación.

En la figura 8.1 se visualizan los porcentajes de los casos de usos implementados en la aplicación.

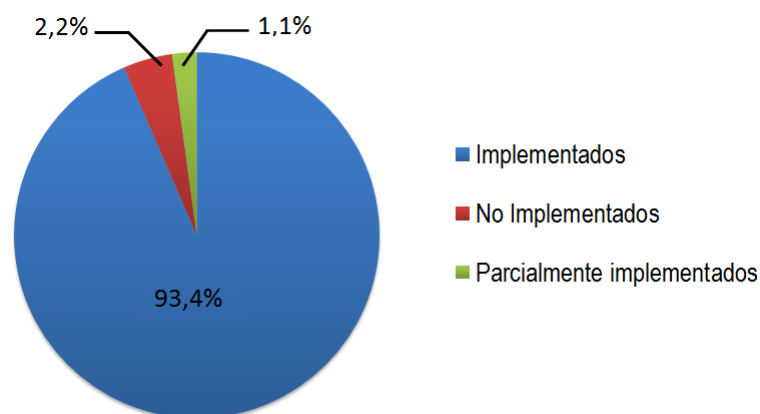


Figura 8.1: Porcentajes de implementación de casos de uso.

Capítulo 8. Evaluación

A partir de este gráfico se observa que de un total de 46 casos definidos se implementaron 43, uno fue parcialmente implementado y dos casos de uso quedaron como trabajo a futuro.

Los casos de uso no implementados son los que describen la forma de brindar ayuda a los usuarios de manera dinámica.

El caso de uso parcialmente desarrollado es “Editar texto de referencia”. La aplicación permite la edición del texto de referencia pero sin integrar la funcionalidad del botón (“LIBRE/OCUPADO”) para resolver la concurrencia de usuarios al momento de la elaboración del texto de referencia.

Por lo que se tiene un alto porcentaje de casos de usos implementados que permite afirmar que en la etapa de la implementación se trabajó de forma de reflejar el análisis propuesto.

8.3. Evaluación a partir de los criterios de éxitos definidos.

Por último es importante una vez cumplido el tiempo de desarrollo del proyecto rever los criterios de éxitos propuestos en un principio. Estos fueron los siguientes:

1. El proyecto alcanza su objetivo antes del 30 de setiembre de 2014.
2. La aplicación cumple con los requerimientos analizados y establecidos en el sistema KLEAR (especificados por el creador de este).
3. El usuario logra ingresar a la aplicación a través de un identificador y contraseña.
4. La aplicación es capaz de manejar al menos dos excepciones de las más frecuentes e informar al usuario su ocurrencia.

El primer criterio no fue alcanzado, ya que se pidió una prórroga.

El segundo criterio está relacionado con lo evaluado al comienzo, donde se consideró que la aplicación refleja la mayor parte de los casos de usos analizados. Y como existe una relación estrecha entre estos últimos y los requerimientos, se entiende que este criterio se verifica en gran medida.

A su vez los últimos criterios son verificados, ya que el ingreso del usuario a la aplicación está restringido por un identificador y una contraseña. Y la aplicación es capaz de manejar ciertas situaciones en donde su comportamiento natural darían algún tipo de error. Ejemplos de ello, es el manejo de la calificación total, cuando todos los pesos tienen valor cero y la situación en la que el usuario deja sin completar la categoría léxica requerida, en este caso el sistema notifica y permite

8.3. Evaluación a partir de los criterios de éxitos definidos.

que se siga trabajando.

De esta forma queda evaluado el proceso del proyecto que sirven como pilares de las conclusiones.

Capítulo 9

Conclusiones y trabajo futuro.

9.1. Conclusiones.

En primer lugar se puede concluir que se cumplió el objetivo del proyecto logrando desarrollar una aplicación web de soporte para el sistema KLEAR con características de una aplicación para usuario final.

El producto logrado permite realizar las funcionalidades requeridas de forma amena, funcional e intuitiva.

Uno de los principales resultados del proyecto es el Análisis de los Casos Uso. Los mismos por su detalle y formato estándar constituyen una base para la construcción de nuevos sistemas o el mejoramiento de éste.

Fue un gran acierto el trabajo con el framework Web2py pues permitió el desarrollo de la aplicación a partir de los conocimientos básicos que se poseían en el área de programación de software. Su gran comunidad de desarrolladores resultó un consultor continuo que brindó soluciones e ideas para la implementación.

La mayor dificultad fue el manejo del tiempo, éste es uno de los mayores aprendizajes que deja este proyecto: jerarquizar las tareas resulta vital para el cumplimiento de los objetivos. El trabajo continuo junto con la perseverancia son las razones de que el conocimiento adquirido y el progreso del proyecto aumentara significativamente con el paso del tiempo.

Fue la primera vez que las integrantes tuvieron el rol de gestor y ejecutor del proyecto. Esto implicó un primer acercamiento con las responsabilidades que implica el cumplimiento de plazos, la distribución de recursos y la estimación de costos. Como grupo se considera que fue una experiencia enriquecedora tanto en lo personal como en lo profesional. Haber realizado un proyecto de carácter informático complementa satisfactoriamente el perfil de Telecomunicaciones, que hoy en día son áreas que se potencian.

Capítulo 9. Conclusiones y trabajo futuro.

9.2. Trabajo futuro.

Dentro de los trabajos futuros se destaca la necesidad de mejorar la portabilidad de la aplicación.

Con el avance del proyecto, y al ejecutar el mismo se detectaron demoras en la ejecución de algunas funcionalidades. El procesamiento del texto ingresado, la verificación de las reglas sintácticas y generación del grafo son ejemplo de ello. Uno de los motivos puede ser la inicialización que se realiza del paquete NLTK en la importación de los módulos del sistema KLEAR, como por ejemplo `kleditor.py`, `wordlist.py`, `validate.py`, `sentences.py`. Para mejorar el tiempo de respuesta debe analizarse cuantas veces se importan estos módulos y minimizar la cantidad de importaciones realizadas a estas librerías. Si se determina que las demoras disminuyen efectivamente con esta acción, es necesario desarrollar un mecanismo por el cual la importación se realice la primera vez que es requerido y que el resultado quede disponible para el resto de la sesión.

Una forma de resolver la concurrencia de usuarios fue planteada en el análisis detallado de los casos de usos en la sección B.5.1 del apéndice. Se propone el uso de un botón (“LIBRE/OCUPADO”) que avisa de la disponibilidad del texto de referencia para ser editado. Esta situación no fue contemplada por razones de tiempo, por lo que se encuentra en este punto una mejora significativa para incluir en esta aplicación.

La forma de proporcionar ayuda a los usuarios, especificada en los requerimientos y analizada en la sección del apéndice B.9 no fue implementada, en cambio se integraron a la aplicación los manuales de usuario. El acceso a estos se realizó a través de la ayuda del menú lateral. Queda pendiente para mejoras futuras la realización completa de esta funcionalidad para brindar una ayuda dinámica a los usuarios.

Se deben realizar mejoras en el diseño de base de datos según lo expuesto en el capítulo 5 para lograr una mayor eficiencia en el manejo de la información.

Quedan pendientes las correcciones de los bugs encontrados en las funcionalidades implementadas para la administración de usuarios y en acciones que realiza el rol administrador.

Por último, resta la verificación de los casos de uso implementados, mediante el protocolo de testeo indicado en el capítulo 6 para este fin.

Apéndice A

Gestión de Proyecto

A.1. Introducción

Se presenta un resumen de lo planificado al inicio de este proyecto, y se analiza el grado de cumplimiento y las desviaciones. Finalmente se hace una evaluación de la gestión.

A.2. Plan del proyecto KLEARapp.

A.2.1. Resumen.

- **Estudiantes:**
 - Almeyda Natalia, C.I.: 4.086.110-7, natalia.almeyda@gmail.com
 - Deus Camila, C.I.: 4.540.621-9, camideuz@hotmail.com
 - Rivas Alejandra, C.I.: 4.003.118-0, alejandrarrivas@yahoo.com
- **Cliente:**
 - Víctor González
- **Tutor:**
 - Víctor González
- **Fecha prevista de finalización:**
 - 30 de setiembre de 2014
- **Total de hs. a realizar previstas por el grupo de proyecto:**
 - 725 horas
- **Entregables asociados a los correspondientes hitos:**

Apéndice A. Gestión de Proyecto

- Primer hito (15 de febrero de 2014)
 - Desarrollo de una aplicación sencilla en web2py.
 - Elaboración de una lista de las facilidades que se le van a proveer al usuario y de que manera se va a brindar.
 - Análisis del sistema
 - Diseño de arquitectura
 - Modelo de página Web
- Segundo hito (17 de julio de 2014)
 - Interfaz de calidad de producción, que incorpore aspectos de seguridad, soporte de usuarios y versatilidad del sistema.
 - Resultados de la aplicación frente a distintas pruebas de uso.

A.2.2. Descripción del proyecto.

Capítulo 1, sección 1.3

A.2.3. Objetivo general.

Capítulo 1, sección 1.1

A.2.4. Alcance.

El alcance de este proyecto es el desarrollo de una aplicación web. Para la realización de esta aplicación se cuenta con bibliotecas de software desarrolladas previamente en el proyecto KLEAR, las cuales resuelven el tratamiento de textos y la construcción de las representaciones de conocimiento.

A.2.5. Criterios de éxito.

- El proyecto alcanza su objetivo antes del 30 de setiembre de 2014.
- La aplicación cumple con los requerimientos analizados y establecidos en el sistema KLEAR (especificados por el creador de éste).
- El usuario logra ingresar a la aplicación a través de un identificador y contraseña.
- La aplicación es capaz de manejar al menos dos excepciones de las más frecuentes e informar al usuario su ocurrencia.

A.2.6. Actores.

- Equipo de proyecto KLEARapp (estudiantes - tutor).
- Creador del Sistema KLEAR.

A.2. Plan del proyecto KLEARapp.

- Cliente.
- Empresa / organización encargada de la contratación del hosting y dominio.

A.2.7. Supuestos.

- Se cuenta con todas las herramientas de software necesarias para el diseño de la aplicación web.
- Las bibliotecas del sistema KLEAR funcionan correctamente
- No existen cambios en las versiones de los software empleados que obliguen a modificar la aplicación web.
- El equipo de proyecto cuenta con los conocimientos necesarios para cumplir con los requerimientos de diseño gráfico que solicita el cliente.
- Los integrantes del equipo son los mismos en el transcurso del proyecto.
- El tiempo dedicado por los integrantes del grupo es el adecuado.

A.2.8. Restricciones.

- El equipo de proyecto tiene una disponibilidad de 10 horas semanales.
- La aplicación web deberá desarrollarse solamente con recursos de software libre.
- El sistema KLEAR ya está previamente programado en un lenguaje y versión específicos.
- El equipo de proyecto no posee experiencia previa en el desarrollo de aplicaciones web.

A.2.9. Especificación funcional del proyecto.

Los productos a desarrollar son una aplicación web para el soporte del sistema KLEAR y las funcionalidades necesarias para acceder a este sistema desde un navegador, con las características y calidad de una aplicación para usuario final. El tratamiento de textos y la construcción de las representaciones de conocimiento están resueltos, y se dispone de las bibliotecas de software correspondientes, desarrolladas en el proyecto KLEAR.

Este proyecto propone la construcción de una aplicación web (de nombre KLEARapp) apta para el uso por estudiantes, que implemente la construcción de grafos semánticos a partir de textos usando los desarrollos del proyecto KLEAR.

Apéndice A. Gestión de Proyecto

El modelo organizativo a emplear para el diseño del software es el MVC. Python es el lenguaje de programación a utilizar.

La presentación de la aplicación, contará con una interfaz de usuario clara, sencilla de manejar, eficiente y grata para el uso por estudiantes y docentes.

Además se tendrán en cuenta temas de seguridad, versatilidad del sistema y se contará con soporte de usuarios.

A.2.10. Objetivos específicos.

- Aprendizaje del lenguaje de programación Python y del marco de desarrollo web2py (basado en Python).
- Analizar y establecer los requerimientos necesarios de la aplicación.
- Comprensión del paradigma de diseño MVC de ingeniería de software.
- Diseño de una arquitectura que refleje el análisis del sistema y se adapte al modelo MVC planteado.
- Diseño y solución de una interfaz apta para el manejo de un usuario final.
- Aprendizaje de los conocimientos necesarios para la creación de una página web (HTML, SQL).
- Verificación del funcionamiento de la aplicación.
- Elaboración de tutorial de usuario y administrador.
- Documentación del proyecto.

A.2.11. Entregables asociados.

- Desarrollo de una aplicación sencilla en web2py.
- Elaboración de una lista de las facilidades que se le van a proveer al usuario y de qué manera se van a brindar.
- Elaboración de un modelo organizativo apropiado para el diseño del software a desarrollar.
- Un diseño que incluye todos los módulos del sistema que responden frente al análisis y al modelo MVC propuesto.
- Interfaz de calidad de producción, que incorpore aspectos de seguridad, soporte de usuarios y versatilidad del sistema.
- Desarrollo de un modelo de página web.
- Resultados de la aplicación frente a distintas pruebas de uso.

A.2. Plan del proyecto KLEARapp.

- Manual de usuario y de administrador.

- Informe con la documentación del proyecto.

A.2.12. Tareas.

ENTREGABLE		TAREA	PRECEDENCIAS	HRS
Aplicación web2py				
	A	Estudio de bibliografía referida a Python y web2py		100
	B	Diseño de aplicación en web2py	A	12
Elaboración de una lista de facilidades				
	C	Análisis de los requerimientos	-	15
	D	Estudio del sistema KLEAR	-	40
	E	Elaboración de la lista de facilidades	C	6
Análisis				
	F	Casos de usos	E	30
	G	Restricciones	E	30
	H	Otras consideraciones	E	20
Diseño de arquitectura				
	I	Estudio del modelo MVC	-	15
	J	Esquema del modelo MVC	F,G,H,I	30
	K	Utilización del modelo MVC en la aplicación	I,J	30
Modelo página Web				
	L	Estudio de software necesarios	-	24
	M	Definir contenido de esa página web.	-	12

Apéndice A. Gestión de Proyecto

ENTREGABLE		TAREA		SUBTAREA	PRECE- DENCIAS	HRS
Interfaz de calidad de producción						
	N	Desarrollo de la aplicación				
			N1	Escribir el código	B,K,E,L	144
			N2	Pruebas del código	N1	50
Resultados: prueba de uso						
	O	Definir posibles escenarios de prueba			N	6
	P	Probar la aplicación en los escenarios definidos.			O	6
	Q	Documentar y analizar resultados.			P	24
Manuales alumno, docente, administrador						
	R	Recolectar y seleccionar documentación relacionada			E,N	12
	S	Armar esquema del contenido del manual			R	4
	T	Redacción de un borrador del manual.			Q,S	8
	U	Diseño gráfico del manual.			Q,S	5
	V	Redacción definitiva del manual.			T,U	12
Documentación del proyecto.						
	W	Registro mensual de avances			-	6
	X	Familiarizarse con una herramienta de compilación de texto			-	12
	Y	Redacción de un borrador del documento			Q	36
	Z	Elaboración del documento.			Y	36

A.2. Plan del proyecto KLEARapp.

A.2.13. Cronograma detallado del proyecto.

En la figura A.1 se presenta el cronograma de tareas realizado al inicio del proyecto.

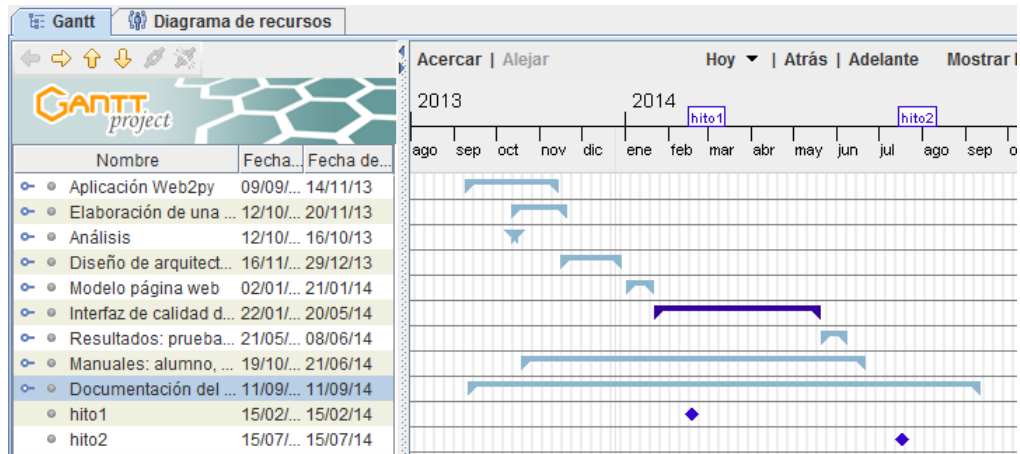


Figura A.1: Diagrama Gantt con el cronograma de tareas.

A.2.14. Análisis de costos.

Para armar el presupuesto se tuvo en cuenta que no se tiene un cliente que pague por este proyecto, por lo que en el flujo de caja sólo se presentaron egresos, que consisten en el costo de las horas hombres dedicadas a la realización del mismo.

A partir del presupuesto elaborado el costo asociado a este proyecto es de U\$S 13.482.

A.2.15. Análisis de Riesgos.

Estimación y evaluación inicial de Riesgos.

Para poder evaluar los riesgos que pueden surgir durante el proceso del proyecto se realizó un análisis de los siguientes ítems:

- Incertidumbres técnicas del Proyecto.
- Supuestos, Actores, Restricciones, son potenciales fuentes de Riesgo.
- Cronograma.

Primero se procedió a listar los posibles riesgos que surgieron luego de la evaluación anterior.

Ellos son:

Apéndice A. Gestión de Proyecto

R1 - No entender el sistema KLEAR.

R2 - El sistema KLEAR no funciona en la forma esperada ó necesita modificaciones.

R3 - Más tiempo requerido en el desarrollo de tareas.

R4 - El tiempo dedicado por los integrantes es insuficiente.

R5 - Falta de experiencia.

R6 - Algún integrante abandona el proyecto.

R7 - Pérdida de información.

R8 - Incompatibilidad de versión de alguna de las herramientas de software requeridas.

Una vez que se cuantificaron los posibles riesgos, se estimó una probabilidad de ocurrencia y un nivel de impacto a cada uno, donde se analizó la vulnerabilidad de los objetivos así como también la viabilidad del Proyecto. A partir de esto se armó la tabla:

A.2. Plan del proyecto KLEARapp.

	RIESGO	PROB. DE OCURRENCIA	IMPACTO
R1	No entender el sistema KLEAR.	MODERADO	EXTREMO
R2	El sistema KLEAR no funciona en la forma esperada ó necesita modificaciones.	POCO PROBABLE	MEDIO
R3	Más tiempo requerido en el desarrollo de tareas.	MODERADO	ALTO
R4	El tiempo dedicado por los integrantes es insuficiente.	POCO PROBABLE	EXTREMO
R5	Falta de experiencia.	MUY PROBABLE	BAJO
R6	Algún integrante abandona el proyecto.	POCO PROBABLE	EXTREMO
R7	Pérdida de información.	POCO PROBABLE	EXTREMO
R8	Incompatibilidad de versión de alguno de las herramientas de software requeridas.	POCO PROBABLE	BAJO

En base a la tabla se construyó la matriz de riesgo, figuraA.2 (probabilidad de ocurrencia vs nivel de impacto).

Plan de respuestas.

Teniendo en cuenta la clasificación de los riesgos en cada uno de los tres niveles, se procedió a elaborar un “plan de respuesta” que contenga acciones que lleven a desplazar los riesgos de alto nivel a uno más bajo, tablaA.1.

	POCO	MODERADO	MUY PROB.
NINGUNO			
BAJO	R8		R5
MEDIO	R2		
ALTO		R3	
EXTREMO	R6 R7 R4	R1	

	- Nivel 1
	- Nivel 2
	- Nivel 3

Figura A.2: Matriz de riesgos.

A.3. Análisis de la gestión realizada.

Los riesgos fueron identificados correctamente, no así la estimación de la probabilidad de ocurrencia y de su impacto. En particular se subestimó dichos factores en los riesgos identificados como R3 y R5. En el riesgo R3 la probabilidad de ocurrencia se debió considerar como “MUY ALTO” y el impacto “EXTREMO”. En el riesgo R5 el impacto debió ser considerado como “ALTO”.

El haberse equivocado en la probabilidad de ocurrencia significó una estima-

A.3. Análisis de la gestión realizada.

RIESGO	TIPO DE RESPUESTA	ACCIÓN
R1	Mitigar el impacto	Hablar con el creador del sistema
R2	Mitigar el impacto	Hablar con el creador del sistema y realizar las modificaciones correspondientes.
R3	Eliminar el riesgo	Control periódico del cumplimiento del cronograma y si es necesario ajustar el mismo.
R4	Mitigar el impacto	Control periódico del cumplimiento del cronograma y si es necesario ajustar el mismo.
R5	Mitigar el impacto	Considerar buffers de tiempo en aquellas tareas que se ven afectadas por la falta de experiencia.
R6	Mitigar el impacto	Tener un respaldo común de la información, comentar el código generado, llevar registro de las tareas y que el conocimiento base del proyecto sea estudiado por todos los integrantes del proyecto.
R7	Mitigar el impacto	Respaldar periódicamente en diferentes fuentes.
R8	Aceptar el riesgo	

Tabla A.1: Acciones de respuesta ante los riesgos.

ción errónea en los tiempos de cada tarea. No acertar en el impacto de cada riesgo tuvo la consecuencia de un cálculo erróneo en los buffer de tiempo extra.

Como consecuencia se recurrió a la solicitud de prórroga un mes antes de la finalización del proyecto.

Apéndice B

Análisis de casos de uso por áreas

B.1. Administración de usuarios

Para ingresar al sistema KLEARapp es necesario estar registrado. Al registrarse, el usuario va a disponer de un nombre de usuario que será su CI y una contraseña, que le permitirán acceder a las funcionalidades del mismo según su rol alumno o docente. La gestión de usuarios de los cursos no tienen vínculo entre ellos. El alumno se matriculará en cada curso. Se presentan distintos escenarios para la administración de usuarios de acuerdo al rol del usuario.

En caso de ser docente también existe una distinción que radica en que si desea registrarse al sistema pero no vinculado a ningún curso existente o por el contrario desea hacerlo como docente de un curso creado previamente.

B.1.1. Registrarse en la aplicación Klearapp con rol alumno.

El alumno desea registrarse en la aplicación para poder participar luego de los cursos.

Descripción completa

Caso de uso: RegistrarseEnLaAplicaciónKlearappConRolAlumno

Actor principal: Alumno

Personal involucrado e intereses:

- ALUMNO: se registra en la aplicación.
- SISTEMA: guarda en la base de datos el nuevo registro.

Precondiciones: el sistema KLEARapp está funcionando correctamente y listo para recibir registros.

Apéndice B. Análisis de casos de uso por áreas

Postcondiciones: el alumno está registrado en la aplicación y podrá iniciar sesión.

Flujo básico:

ACTOR		SISTEMA	
1	El Alumno hace clic en el botón “ALUMNO” de la pantalla principal.	2	Muestra la pantalla de iniciar sesión.
3	El Alumno selecciona “REGÍSTRESE”.	4	Despliega un formulario “ficha de matriculación” con los campos obligatorios para registrarse en la aplicación.
5	El Alumno completa la “ficha de matriculación” (Nombre, Apellido, C.I., correo electrónico, contraseña). Luego hace clic en “REGÍSTRESE”.	6	Verifica que no exista la cédula de identidad ingresada en el sistema ya sea como Alumno, Docente u Administrador.
		7	Si la cédula no está registrada, el sistema guarda en la base de datos el nuevo usuario.

Flujo alternativo:

	7a2: Ingresa su cédula nuevamente.	7a	La cédula está registrada. 7a1: Despliega un mensaje indicado que ya existe esa cédula en la base de datos.
	7a2: Ingresa su cédula nuevamente.	7b	La cédula no tiene el formato correcto. 7a1: Despliega un mensaje indicado que ingrese su cédula sin el dígito verificador.

Tabla B.1: Registrarse en la aplicación con rol alumno.

B.1.2. Registrarse en la aplicación Klearapp con rol docente.

El docente desea registrarse en la aplicación para poder crear y administrar sus cursos.

Descripción completa

Caso de uso: RegistrarseEnLaAplicaciónKlearappConRolDocente

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** se registra en la aplicación.
- **SISTEMA:** guarda en la base de datos el nuevo registro.

Precondiciones: el sistema KLEARapp está funcionando correctamente y listo para recibir registros.

Postcondiciones: el docente está registrado en la aplicación y podrá iniciar sesión.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón “DOCENTE” de la pantalla principal.	2	Muestra la pantalla de iniciar sesión.
3	El Docente selecciona “REGÍSTRESE”.	4	Despliega un formulario “ficha de matriculación” con los campos obligatorios para registrarse en la aplicación.
5	El Docente completa la “ficha de matriculación” (Nombre, Apellido, C.I., correo electrónico, contraseña). Luego hace clic en “REGÍSTRESE”.	6	Verifica que no exista la cédula de identidad ingresada en el sistema ya sea como Alumno, Docente u Administrador.
		7	Si la cédula no está registrada, el sistema guarda en la base de datos el nuevo usuario.

Apéndice B. Análisis de casos de uso por áreas

Flujo alternativo:

	7a2: Ingresa su cédula nuevamente.	7a	La cédula está registrada. 7a1: Despliega un mensaje indicado que ya existe esa cédula en la base de datos.
	7a2: Ingresa su cédula nuevamente.	7b	La cédula no tiene el formato correcto. 7a1: Despliega un mensaje indicado que ingrese su cédula sin el dígito verificador.

Tabla B.2: Registrarse en la aplicación con rol docente.

B.1.3. Registrarse en la aplicación Klearapp con rol administrador.

El administrador desea registrarse en la aplicación para poder administrar a los usuarios con rol docente.

Descripción completa

Caso de uso: RegistrarseEnLaAplicaciónKlearappConRolAdministrador

Actor principal: Administrador

Personal involucrado e intereses:

- ADMINISTRADOR: se registra en la aplicación.
- SISTEMA: guarda en la base de datos el nuevo registro.

Precondiciones: el sistema KLEARapp está funcionando correctamente y listo para recibir registros.

Postcondiciones: el administrador está registrado en la aplicación y podrá iniciar sesión.

B.1. Administración de usuarios

Flujo básico:

ACTOR		SISTEMA	
1	El Administrador hace clic en el botón “DOCENTE” o “ALUMNO” de la pantalla principal.	2	Muestra la pantalla de iniciar sesión.
3	El Administrador selecciona “REGÍSTRESE”.	4	Despliega un formulario “ficha de matriculación” con los campos obligatorios para registrarse en la aplicación.
5	El Administrador completa la “ficha de matriculación” con la salvedad que debe poner en el campo Nombre: ‘Admin.klear2014’, en el resto de los campos (Apellido, C.I., correo electrónico, contraseña)debe poner su información personal. Luego hace clic en “REGÍSTRESE”.	6	Verifica que no exista la cédula de identidad ingresada ya sea como Alumno, Docente u Administrador.
		7	Si la cédula no está registrada, el sistema guarda en la base de datos el nuevo usuario.

Flujo alternativo:

	7a2: Ingresar su cédula nuevamente.	7a	La cédula está registrada. 7a1: Despliega un mensaje indicado que ya existe esa cédula en la base de datos.
	7a2: Ingresar su cédula nuevamente.	7b	La cédula no tiene el formato correcto. 7a1: Despliega un mensaje indicado que ingrese su cédula sin el dígito verificador.

Tabla B.3: Registrarse en la aplicación con rol administrador.

Apéndice B. Análisis de casos de uso por áreas

B.1.4. Solicitar matriculación a un curso.

El Alumno desea matricularse a un curso existente para poder participar de las actividades del mismo. Para ello a través del sistema solicita permiso.

Descripción completa

Caso de uso: SolicitarMatriculacionAUnCurso

Actor principal: Alumno

Personal involucrado e intereses:

- ALUMNO: solicita matricularse en el curso correspondiente.
- SISTEMA: indica solicitud a los docentes.

Precondiciones: el alumno ya está registrado y ha iniciado sesión.

Postcondiciones: el docente tiene la solicitud pendiente para decidir aceptar o no al alumno en su curso.

Flujo básico:

ACTOR		SISTEMA	
1	El Alumno hace clic en el curso que desea matricularse.	2	Verifica que el Alumno no haya solicitado matriculación antes. Si es la primera vez que lo solicita, muestra la pantalla preguntando si desea matricularse y las opciones: 'SI' 'NO' .
3	El Alumno selecciona "SI" .	4	Retorna a la pantalla de la lista de cursos. Despliega un mensaje emergente avisando que la solicitud ha sido enviada con éxito. El sistema indica al docente esta solicitud de matriculación pendiente y guarda en la base de datos el usuario en estado "PENDIENTE"

Flujo alternativo:

		2a	<p>Detecta que ya ha pedido matriculación en ese curso.</p> <p>2a1:Despliega un mensaje emergente avisando que la solicitud ya fue realizada y permanece en la pantalla de la lista de cursos.</p>
3a	<p>El Alumno no desea matricularse al curso seleccionado.</p> <p>3a1:Hace clic en “NO”.</p>		<p>3a2:Retorna a la pantalla con la lista de cursos.</p>

Tabla B.4: Solicitar matriculación a un curso.

B.1.5. Confirmar matriculación de Alumno.

El docente es quien decide aceptar o no la solicitud de matriculación del alumno al curso dependiendo de las condiciones para el curso en particular.

Descripción completa

Caso de uso: ConfirmarMatriculaciónDeAlumno

Actor principal: Docente

Personal involucrado e intereses:

- DOCENTE: confirma solicitud del alumno.
- SISTEMA: ingresa al alumno como usuario.

Precondiciones: el docente tiene pendiente una nueva solicitud de matriculación.

Postcondiciones: el alumno recibe por mail la confirmación para ingresar al curso.

Apéndice B. Análisis de casos de uso por áreas

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón “SOLICITUDES” en la barra lateral dentro del curso.	2	Muestra la pantalla con la lista de solicitudes pendientes correspondiente a los alumnos.
3	El Docente cambia el estado de los alumnos que desea aceptar, pasando de “PENDIENTE” a “ACTIVO” y hace clic en el botón “ENVIAR”.	4	Cancela solicitud pendiente y cambia el estado
5	El Docente hace clic en el botón “CURSO ACTUAL” de la barra lateral del curso y envía por mail (por su cuenta) la confirmación al alumno.	6	Vuelve a la pantalla principal del curso.

Flujo alternativo:

3a	<p>El usuario no cumple con los requisitos para estar matriculado en el curso.</p> <p>3a1:El Docente rechaza la solicitud pasando el estado de “PENDIENTE” a “BORRADO”.</p> <p>3a3:hace clic en el botón “CURSO ACTUAL” de la barra lateral del curso. El docente envía mail explicando al usuario por qué no cumple con los requisitos.</p>		<p>3a2: Borra esta solicitud pendiente y cambia el estado del alumno a “BORRADO”.</p>
-----------	--	--	---

Tabla B.5: Confirmar matriculación de alumno.

B.1.6. Eliminar usuario Alumno de un curso.

El Alumno desea borrarse de un curso.

Descripción completa

Caso de uso: SolicitarEliminacionUsuarioAlumnoDeCurso

Actor principal:Alumno

Personal involucrado e intereses:

- ALUMNO: se des matricula de un curso.
- SISTEMA: cambia en la base de datos el estado que tiene el alumno en ese curso. (Pasa de “ACTIVO” a “BORRADO”).

Precondiciones: el alumno cuenta con un usuario y se encuentra matriculado en el curso del que desea borrarse y se ha autenticado con su usuario.

Postcondiciones: en la base de datos el estado que tiene el alumno en ese curso es “BORRADO”.

Flujo básico:

ACTOR		SISTEMA	
1	El Alumno hace clic en el botón “DESMATRICULARME” del menú lateral del curso.	2	Muestra la pantalla preguntándole al Alumno si desea desmatricularse y las opciones: ‘SI’ ‘NO’ .
3	El Alumno confirma que desea borrarse del curso haciendo clic en “SI”.	4	Retorna a la pantalla con la lista de cursos, cambia en la base de datos el estado que tiene el Alumno en ese curso (pasa de “ACTIVO” a “BORRADO”) e inicia el temporizador que tiene como valor el período de restablecimiento (siete días). Además despliega un mensaje indicado que la solicitud de desmatriculación se envió exitosamente.
		5	Cumplido el período de restablecimiento y sin haber recibido ninguna solicitud, borra toda la información asociada a ese Alumno en el curso sin posibilidad de recuperarla.

Apéndice B. Análisis de casos de uso por áreas

Flujo alternativo:

3a	El Alumno no desea desmatricularse al curso seleccionado. 3a1:Hace clic en “NO”.		3a2:Retorna a la pantalla del curso.
-----------	---	--	--------------------------------------

Tabla B.6: Desmatriculación de alumno de un curso.

B.1.7. Eliminación de usuario Alumno por parte de un docente.

El docente desea borrar un alumno de su curso.

Descripción completa

Caso de uso: EliminaciónDeUsuarioAlumnoPorParteDeUnDocente

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** des matricula a un alumno de su curso.
- **SISTEMA:** cambia en la base de datos el estado que tiene el alumno en ese curso. (Pasa de “ACTIVO” a “BORRADO”).

Precondiciones: el alumno que se desea desmatricular del curso cuenta con un usuario en éste. El docente se ha registrado en el sistema e ingresó al curso del que desea eliminar un alumno.

Postcondiciones: en la base de datos el estado que tiene el alumno en ese curso es “BORRADO”.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón “LISTA DE ALUMNOS” de la pantalla del curso.	2	Despliega una pantalla mostrando la lista de alumnos del curso con sus datos, incluido el estado.
3	El Docente selecciona el o los alumnos que desea eliminar del curso. Luego hace clic en “DESMATRICULAR”.	4	Cambia en la base de datos el estado que tiene/n el/los alumno/s en ese curso (pasando de “ACTIVO” a “BORRADO”), refleja dicho cambio en la lista de alumnos que ven los Docentes del curso e inicia el temporizador que tiene como valor el período de restablecimiento (siete días).
		5	Cumplido el período de restablecimiento y sin haber recibido ninguna solicitud, borra toda la información asociada a ese alumno en el curso sin posibilidad de recuperarla.

Tabla B.7: Eliminación de alumno por parte de un docente.

B.1.8. Deshacer eliminación de usuario Alumno por parte de un docente.

El docente desea reestablecer a un alumno de su curso que antes fue borrado ya sea por un docente o por él mismo.

Descripción completa

Caso de uso: DeshacerEliminacionDeUsuarioAlumnoPorParteDeUnDocente

Actor principal: Docente

Personal involucrado e intereses:

- DOCENTE: vuelve a activar a un alumno de su curso.

Apéndice B. Análisis de casos de uso por áreas

- **SISTEMA:** cambia en la base de datos el estado que tiene el alumno en ese curso, pasa de “BORRADO” a “ACTIVO”.

Precondiciones: el alumno que se desea reestablecer en el curso ha sido borrado de éste y está dentro del tiempo de reestablecimiento . El docente se ha registrado en el sistema e ingresó al curso del que desea reactivar un alumno.

Postcondiciones: en la base de datos el estado que tiene el alumno en ese curso es “ACTIVO”.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón “REACTIVACIÓN” en la barra lateral del curso.	2	Despliega una pantalla mostrando la lista de alumnos del curso cuyo estado es “BORRADO”.
3	El Docente hace clic en el botón “EDITAR” en la fila del alumno a reactivar.	4	El sistema muestra en pantalla los datos del alumno.
5	El Docente cambia el estado pasando de “BORRADO” a “ACTIVO” y hace clic en el botón “ENVIAR”	6	Cambia en la base de datos el estado que tiene el alumno en ese curso, pasando de “BORRADO” a “ACTIVO”, y refleja dicho cambio en la lista de alumnos que ven los Docentes del curso. Además vuelve a cero el temporizador con el tiempo de reestablecimiento.
7	Envía por mail (por su cuenta) la notificación de reactivación.		

Tabla B.8: Deshacer eliminación de alumno por parte de un docente.

B.1.9. Solicitar matriculación con rol docente en curso existente.

El usuario en rol de docente que se llamará Docente B desea matricularse en un curso creado anteriormente por otro docente, el que se designará como Docente A. Para ello se matricula en el curso, esta solicitud es vista por el Docente A y el Administrador.

Descripción completa

Caso de uso: MatricularDocenteEnCursoExistente

Actor principal: Docente B

Personal involucrado e intereses:

- **DOCENTE:** solicita matricularse en el curso correspondiente.
- **SISTEMA:** indica solicitud a los docentes.

Precondiciones: el docente ya está registrado y ha iniciado sesión.

Postcondiciones: el docente A tiene la solicitud pendiente para decidir aceptar o no al docente B en su curso.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente B hace clic en el curso que desea matricularse.	2	Verifica que el Docente B no haya solicitado matriculación antes. Si es la primera vez que lo solicita, muestra la pantalla preguntando si desea matricularse y las opciones: 'SI' 'NO' .
3	El Docente B selecciona 'SI'.	4	Retorna a la pantalla de la lista de cursos. Despliega un mensaje emergente avisando que la solicitud ha sido enviada con éxito. El sistema indica al docente A esta solicitud de matriculación pendiente y guarda en la base de datos el usuario en estado "PENDIENTE"

Apéndice B. Análisis de casos de uso por áreas

Flujo alternativo:

		2a	Detecta que ya ha pedido matriculación en ese curso. 2a1:Despliega un mensaje emergente avisando que la solicitud ya fue realizada y permanece en la pantalla de la lista de cursos.
3a	El Docente B no desea matricularse al curso seleccionado. 3a1:Hace clic en “NO”.		3a2:Retorna a la pantalla con la lista de cursos.

Tabla B.9: Solicitar matriculación con rol docente en curso existente.

B.1.10. Confirmar matriculación de Docente.

El Docente B sólo podrá ingresar al curso a partir del momento en que el Docente A o el Administrador acepten esta solicitud. Éste último sólo responde como último recurso.

Descripción completa

Caso de uso: ConfirmarMatriculaciónDeDocente

Actor principal: Docente A

Personal involucrado e intereses:

- DOCENTE A: confirma solicitud del Docente B.
- SISTEMA: ingresa al docente como docente del curso.

Precondiciones: el docente tiene pendiente una nueva solicitud de matriculación.

Postcondiciones: el docente B recibe por mail la confirmación para ingresar al curso.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente A hace clic en el botón “SOLICITUDES” en la barra lateral dentro del curso.	2	Muestra la pantalla con la lista de solicitudes pendientes correspondiente a los docentes.
3	El Docente A cambia el estado de los docentes B que desea aceptar, pasando de “PENDIENTE” a “ACTIVO” y hace clic en el botón “ENVIAR”.	4	Cancela solicitud pendiente y cambia los datos del Docente B en su base de datos. Dándole permiso para ingresar y editar en el curso en el que se matriculó.
5	El Docente A hace clic en el botón “CURSO ACTUAL” del menú lateral del curso y envía por mail (por su cuenta) la confirmación al docente B.	6	Vuelve a la pantalla principal del curso.

Flujo alternativo:

3a	<p>El Docente A rechaza la solicitud por parte del DOCENTE B.</p> <p>3a1:El Docente A rechaza la solicitud pasando el estado de “PENDIENTE” a “BORRADO”.</p> <p>3a3:hace clic en el botón “CURSO ACTUAL” del menú lateral del curso. El docente envía mail por su cuenta al docente B explicando el motivo.</p>		<p>3a2: Borra esta solicitud pendiente y cambia el estado del docente B a “BORRADO”.</p>
-----------	---	--	--

Tabla B.10: Confirmar matriculación de Docente.

Apéndice B. Análisis de casos de uso por áreas

B.1.11. Eliminar usuario docente de un curso.

El docente desea no seguir vinculado a un curso.

Descripción completa

Caso de uso: EliminarUsuarioDocenteDeUnCurso

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** dejar de ser docente de un curso.
- **SISTEMA:** cambia en la base de datos el estado que tiene el docente en ese curso, y refleja dicho cambio en la lista de docentes que ve el administrador y en la lista de docentes que ven los otros docentes de ese curso (si existieran).

Precondiciones: el docente se encuentra matriculado como docente en el curso del que desea borrarse y se ha autenticado con su usuario.

Postcondiciones: cambia en la base de datos el estado que tiene el docente en ese curso a “BORRADO”.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón “DESMATRICULARME” del menú lateral del curso.	2	Muestra la pantalla preguntándole al Docente si desea desmatricularse y las opciones: ‘SI’ ‘NO’ .
3	El Docente confirma que desea borrarse del curso haciendo clic en “SI”.	4	Retorna a la pantalla con la lista de cursos, cambia en la base de datos el estado que tiene el Docente en ese curso (pasa de “ACTIVO” a “BORRADO”) e inicia el temporizador que tiene como valor el período de restablecimiento (siete días). Además despliega un mensaje indicado que la solicitud de desmatriculación se envió exitosamente.
		5	Cumplido el período de restablecimiento y sin haber recibido ninguna solicitud, borra toda la información asociada a ese Docente en el curso sin posibilidad de recuperarla.

Flujo alternativo:

3a	<p>El Docente no desea desmatricularse al curso seleccionado.</p> <p>3a1:Hace clic en “NO”.</p>		<p>3a2:Retorna a la pantalla del curso.</p>
-----------	---	--	---

Tabla B.11: Eliminar usuario docente de un curso.

B.1.12. Restablecer usuario docente a un curso por parte de otro docente del curso.

El docente que llamaremos Docente B se ha eliminado de un curso. El Docente A que también pertenece a dicho curso, será quién lo restablezca. El período de siete días para poder restablecer la matrícula en el curso es conocido, tanto por los docentes como por el administrador.

Descripción completa

Caso de uso: RestablecerUsuarioDocenteAUnCursoPorParteDeOtroDocenteDelCurso.

Actor principal: Docente A

Personal involucrado e intereses:

- **DOCENTE A:** restablece la matrícula del DocenteB.
- **SISTEMA:** cambia en la base de datos el estado que tiene el docente en ese curso, y refleja dicho cambio en la lista de docentes que ve el administrador y en la lista de docentes que ven los otros docentes y alumnos de ese curso.

Precondiciones: el DocenteB ha sido desmatriculado recientemente (dentro de un período establecido por la aplicación), cuenta con un usuario en el sistema.

Postcondiciones: en la base de datos el estado que tiene el docente en ese curso es “ACTIVO”.

Apéndice B. Análisis de casos de uso por áreas

Flujo básico:

ACTOR		SISTEMA	
1	El Docente A hace clic en el botón “REACTIVACIÓN” en la barra lateral del curso.	2	Despliega una pantalla mostrando la lista de docentes del curso cuyo estado es “BORRADO”.
3	El Docente A hace clic en el botón “EDITAR” en la fila del docente B para reestablecerle la matrícula del curso.	4	El sistema muestra en pantalla los datos del docente.
5	El Docente A cambia el estado pasando de “BORRADO” a “ACTIVO” y hace clic en el botón “ENVIAR”	6	Cambia en la base de datos el estado que tiene el docente B en ese curso, pasando de “BORRADO” a “ACTIVO”, y refleja dicho cambio en la lista de docentes que ven los Docentes del curso. Además vuelve a cero el temporizador con el tiempo de reestablecimiento.
7	Envía por mail (por su cuenta) la notificación de reactivación.		

Tabla B.12: Restablecer docente por otro docente del curso.

B.1.13. Restablecer matriculación de usuario docente a un curso por parte del administrador.

El único docente que tenía el curso se ha eliminado de éste y es necesario restablecerse su matriculación. Debido a que no existen otros docentes en el curso que realicen ésta tarea, la responsabilidad recae sobre el administrador.

El período para poder restablecer la matrícula en el curso es un tiempo conocido, tanto por los docentes como por el administrador.

Descripción completa

Caso de uso: RestablecerMatriculaciónDeUsuarioDocenteAUnCursoPorParteDelAdministrador.

Actor principal: Administrador

Personal involucrado e intereses:

- ADMINISTRADOR: restablece la matrícula del docente.
- SISTEMA: cambia en la base de datos el estado que tiene el docente en ese curso, y refleja dicho cambio en la lista de docentes que ve el administrador.

Precondiciones: el docente ha sido desmatriculado recientemente (dentro de un período establecido por la aplicación) y cuenta con un usuario en el sistema.

Postcondiciones: en la base de datos el estado que tiene el docente en ese curso es “ACTIVO”.

Flujo básico:

ACTOR		SISTEMA	
1	El Administrador hace clic en el botón “LISTA DOCENTES” del menú lateral de su pantalla principal.	2	Despliega una pantalla mostrando la lista de todos los docentes, cursos y estados.
3	El Administrador hace clic en el botón “EDITAR” en la fila del docente para reestablecerle la matrícula del curso.	4	El sistema muestra en pantalla los datos del docente.
5	El Administrador cambia el estado pasando de “BORRADO” a “ACTIVO” y borra el tiempo de reestablecimiento, luego hace clic en el botón “ENVIAR”	6	Cambia en la base de datos el estado que tiene el docente en ese curso, pasando de “BORRADO” a “ACTIVO”. Además vuelve a cero el temporizador con el tiempo de reestablecimiento.
7	Envía por mail (por su cuenta) la notificación de reactivación.		

Tabla B.13: Restablecer matriculación de docente a un curso por parte del administrador.

Apéndice B. Análisis de casos de uso por áreas

B.1.14. Eliminación de usuario docente de todos los cursos por parte del administrador.

El administrador debe borrar a un docente de todos los cursos.

Descripción completa

Caso de uso: EliminacionDeUsuarioDocnteDeTodoslosCursosPorParteDelAdministrador.

Actor principal: Administrador

Personal involucrado e intereses:

- **ADMINISTRADOR:** borra a docente de todos los cursos.
- **SISTEMA:** cambia en la base de datos el estado que tiene el docente en todos los cursos en los que él es docente, y refleja dicho cambio en la lista de docentes que ve el administrador.

Precondiciones: el docente debe ser borrado de los cursos.

Postcondiciones: en la base de datos el estado que tiene el docente es “BORRADO”.

Flujo básico:

ACTOR		SISTEMA	
1	El Administrador hace clic en el botón “LISTA DOCENTES” del menú lateral de su pantalla principal.	2	Despliega una pantalla mostrando la lista de todos los docentes, cursos y estados.
3	El Administrador hace clic en el botón “EDITAR” en la fila del docente.	4	El sistema muestra en pantalla los datos del docente.
5	El Administrador cambia el estado pasando de “ACTIVO” a “BORRADO”, luego hace clic en el botón “ENVIAR”	6	Cambia en la base de datos el estado que tiene el docente en todos los cursos, pasando de activo a borrado, y refleja dicho cambio en la lista de docentes que ven los Docentes del curso.

Tabla B.14: Eliminación de docente de todos los cursos por parte del administrador.

B.2. Administración de Cursos

Los cursos dividen las grandes áreas temáticas de conocimiento.

En cada uno de ellos se podrá encontrar una serie de unidades de aprendizaje que refieren a un tema más particular dentro de esta área general.

En cada curso participan los docentes quienes son los encargados de la creación y administración de los mismos y los alumnos que lo utilizan como herramienta de aprendizaje.

B.2.1. Crear un nuevo curso en el sistema KLEARapp.

El docente crea un nuevo curso en el sistema (es el único usuario con privilegios que le permiten hacerlo).

Descripción completa

Caso de uso: CrearUnNuevoCursoEnElSistemaKLEARapp.

Actor principal: Docente

Personal involucrado e intereses:

- DOCENTE: crea un nuevo curso.
- SISTEMA: genera y guarda el nuevo curso.

Precondiciones: el docente ya está registrado en el sistema KLEARapp.

Postcondiciones: el curso es visible por todos los usuarios luego de iniciar sesión.

Apéndice B. Análisis de casos de uso por áreas

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón “AGREGAR CURSO” de la pantalla con la lista de cursos.	2	Despliega una pantalla para generar el nuevo curso con campos a completar (nombre de curso y descripción).
3	El Docente completa los campos y hace clic en botón “ENVIAR”.	4	Guarda la configuración, matricula automáticamente al docente que crea el curso en ese curso y muestra la pantalla del nuevo curso, (la que contiene el menú en el lateral izquierdo).

Tabla B.15: Crear un nuevo curso en el sistema KLEARapp.

B.2.2. Borrar un curso del sistema KLEARapp por parte de un docente.

El docente borra un curso del sistema.

Descripción completa

Caso de uso: BorrarUnCursoDelSistemaKLEARappPorParteDeUnDocente.

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** realiza los pasos pertinentes para la eliminación de un curso del sistema.
- **SISTEMA:** elimina el curso.

Precondiciones: el docente ya está registrado en el sistema KLEARapp y es docente del curso que quiere eliminar.

Postcondiciones: el curso deja de ser visible por todos los usuarios matriculados en ese curso.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón “ELIMINAR CURSO” del menú de la pantalla del cursos.	2	Despliega una pantalla en la que le permite al docente marcar un casillero para eliminar.
3	El Docente marca la casilla.	4	Despliega un mensaje preguntándole si está seguro de realizar esa acción.
5	El Docente hace clic en el botón “ACEPTAR” y en “ENVIAR”.	6	Cierra el mensaje, elimina el curso, y actualiza los cursos que se listan en la pantalla Además despliega un mensaje emergente que avisa al Docente que el curso ha sido borrado exitosamente.

Flujo alternativo:

3a	El Docente no desea eliminar al curso seleccionado. 3a1:Hace clic en “CANCELAR”.		
----	---	--	--

Tabla B.16: Borrar un curso del sistema KLEARapp por parte de un docente.

B.2.3. Modificar curso.

El docente desea cambiar alguno de los campos del curso.

Descripción completa

Caso de uso: ModificarCamposDeCursoDelSistemaKLEARappPorParteDeUnDocente.

Actor principal: Docente

Personal involucrado e intereses:

- DOCENTE: realiza los pasos pertinentes para modificar algún campo del curso.
- SISTEMA: modifica campos del curso.

Apéndice B. Análisis de casos de uso por áreas

Precondiciones: el docente ya está registrado en el sistema KLEARapp y el curso ha sido previamente creado.

Postcondiciones: se modificó algún campo del curso.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón “MODIFICAR CURSO” del menú de la pantalla del cursos.	2	Despliega una pantalla en la que le permite al docente editar los campos.
3	El Docente modifica los campos deseados y hace clic en “ACTUALIZAR”.	4	El sistema guarda en su base de datos las modificaciones realizadas y retorna a la pantalla del curso.

Flujo alternativo:

3a	El Docente no desea guardar las modificaciones. 3a1:Hace clic en “CURSO ACTUAL” en el menú lateral.		3a2:Retorna a la pantalla del curso.
----	--	--	--------------------------------------

Tabla B.17: Modificar curso.

B.2.4. Borrar un curso del sistema KLEARapp por parte del administrador.

El administrador borra un curso del sistema. Esto se realizará solo en el caso de no disponer de un docente del curso que lleve a cabo esta tarea.

Descripción completa

Caso de uso: BorrarUnCursoDelSistemaKLEARappPorParteDelAdministrador.

Actor principal: Administrador

Personal involucrado e intereses:

B.2. Administración de Cursos

- ADMINISTRADOR: realiza los pasos pertinentes para la eliminación de un curso del sistema.
- SISTEMA: elimina el curso.

Precondiciones: el Administrador recibió la orden (por fuera) de eliminar un curso del Sistema KLEARapp.

Postcondiciones: el curso deja de ser visible por todos los usuarios matriculados en ese curso.

Flujo básico:

ACTOR		SISTEMA	
1	El Administrador hace clic en el botón “ELIMINAR CURSOS” de la pantalla principal.	2	Despliega una pantalla en la que le permite al administrador eliminar el curso.
3	El Administrador hace clic en el botón “ELIMINAR” de la fila del curso para eliminar.	4	Despliega un mensaje preguntándole si está seguro de realizar esa acción.
5	El Administrador hace clic en “ACEPTAR”.	6	Elimina el curso y actualiza los cursos que se listan en la pantalla.

Flujo alternativo:

5a	El Administrador no desea borrar el cursos seleccionado. 5a1: El Administrador selecciona “CANCELAR”		5a2: Cierra la ventana emergente y vuelve a la lista de cursos sin producirse ningún cambio.
----	---	--	--

Tabla B.18: Borrar un curso del sistema KLEARapp por parte del administrador.

Apéndice B. Análisis de casos de uso por áreas

B.2.5. Usuario ingresa al curso.

Una vez que el usuario se matricula y es aceptado por el docente del curso, podrá ingresar al mismo. Para ingresar a un curso en particular, deberá hacer clic sobre el link del curso.

Descripción completa

Caso de uso: UsuarioIngresaAlCurso.

Actor principal: Usuario

Personal involucrado e intereses:

- USUARIO: visualiza los cursos e ingresa a uno en particular.
- SISTEMA: muestra los cursos, muestra en pantalla el curso seleccionado.

Precondiciones: el usuario ha sido previamente matriculado en algún curso.

Postcondiciones: el usuario ingresa al curso seleccionado.

Flujo básico:

ACTOR		SISTEMA	
1	El usuario inicia sesión.	2	Despliega una pantalla la lista de todos los cursos.
3	El Usuario hace clic en el curso al que desea ingresar.	4	El sistema muestra la pantalla principal del curso.

Flujo alternativo:

3a	El Usuario aún no fue aceptado en el curso 3a1: El Usuario hace clic en el curso al que desea ingresar.		3a2: Despliega un mensaje que explica la situación por la que no puede ingresar.
----	--	--	--

Tabla B.19: Usuario ingresa al curso.

B.3. Administración de Unidad de Aprendizaje (U.A.)

Una unidad de aprendizaje es un bloque que reúne a todos los recursos (tareas, evaluaciones, textos de referencia) necesarios para cumplir con un objetivo de aprendizaje.

El docente es quien administra los bloques: tiene permiso para crear y quitar una o varias U.A.. A su vez puede agregar tareas y evaluaciones a los mismos.

Una vez que el docente crea una U.A. debe especificar una lista de campos (título, descripción, subtema, etc.) referidos al tema al que concierne el bloque.

Cada U.A. es visible también para los alumnos, quienes podrán acceder a las tareas y evaluaciones de las mismas.

B.3.1. Crear una U.A.

En este caso de uso, el docente crea un bloque correspondiente a una U.A. Luego el sistema lo muestra en el sector central de la aplicación.

Dentro del bloque de la U.A. se visualizan varios botones, ellos permiten crear tareas y evaluaciones.

Descripción completa

Caso de uso: CrearUnaU.A.

Actor principal: Docente

Personal involucrado e intereses:

- DOCENTE: crea una U.A.
- SISTEMA: genera y muestra el bloque U.A.

Precondiciones: el curso al que corresponde la U.A. que se quiere crear ya está activo. El docente ya está matriculado en el curso.

Postcondiciones: el sistema mostrará el bloque de la nueva U.A. dentro del espacio del curso correspondiente.

Apéndice B. Análisis de casos de uso por áreas

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón “CREAR U.A.” en el menú lateral izquierdo. de la pantalla con la lista de cursos.	2	Despliega una pantalla para generar la nueva U.A. con campos a completar (nombre de U.A. y descripción).
3	El Docente completa los campos y hace clic en botón “ENVIAR”.	4	Guarda la configuración, y muestra la pantalla del curso con la nueva U.A., y sus atributos.

Tabla B.20: Crear una U.A.

B.3.2. Eliminar una U.A.

El Docente puede eliminar una U.A. específica de uno de sus cursos.

Descripción completa

Caso de uso: EliminarUnaU.A.

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** hace los pasos pertinentes para eliminar una U.A..
- **SISTEMA:** elimina la U.A. y los recursos que contiene.

Precondiciones: la U.A. está creada. El docente ya está matriculado en el curso.

Postcondiciones: en el curso correspondiente se deja de ver la U.A eliminada.

B.3. Administración de Unidad de Aprendizaje (U.A.)

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón “ELIMINAR U.A” en el menú lateral izquierdo.	2	Muestra una pantalla en la que permite al Docente seleccionar cuál de las U.A.’s desea eliminar.
3	El Docente selecciona las U.A.’s que desea borrar y hace clic en “ELIMINAR”.	4	Elimina las U.A.’s y sus atributos, actualiza las que se listan en la pantalla.

Tabla B.21: Eliminar una U.A.

B.3.3. Modificar una U.A.

Una vez creada la U.A. es posible modificar alguno de sus campos.

Descripción completa

Caso de uso: ModificarUnaU.A.

Actor principal: Docente

Personal involucrado e intereses:

- DOCENTE: modifica la U.A. seleccionada.
- SISTEMA: guarda en la base de datos las modificaciones correspondientes.

Precondiciones: la U.A. que se desea modificar fue previamente creada.

Postcondiciones: se guardan las modificaciones realizadas y se muestran en pantalla.

Apéndice B. Análisis de casos de uso por áreas

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón “MODIFICAR U.A” en el menú lateral izquierdo.	2	Muestra una pantalla en la que permite al Docente seleccionar cuál de las U.A.’s desea modificar.
3	El Docente hace clic en la U.A. que desea modificar.	4	Despliega una pantalla en la que le permite al docente editar los campos.
5	El Docente modifica los campos deseados y hace clic en “ACTUALIZAR”.	6	El sistema guarda en su base de datos las modificaciones realizadas y retorna a la pantalla del curso.

Flujo alternativo:

3a	El Docente no desea guardar las modificaciones. 3a1: Hace clic en “CURSO ACTUAL” en el menú lateral.		3a2: Retorna a la pantalla del curso.
-----------	---	--	---------------------------------------

Tabla B.22: Modificar una U.A.

B.4. Manejo de tareas

La tarea de una U.A. contiene el material necesario para alcanzar el objetivo de aprendizaje correspondiente a esa U.A.. Para esto se incluye un espacio donde se puede editar un texto de forma colaborativa por todos los alumnos del curso. A estos textos se los denominan textos de referencia.

B.4.1. Crear una Tarea.

El docente crea una tarea dentro del bloque correspondiente a una U.A.

Descripción completa

Caso de uso: CrearUnaTarea

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** crea una Tarea.
- **SISTEMA:** genera y muestra el link de la tarea dentro de la U.A correspondiente.

Precondiciones: la U.A. a la que pertenece la tarea ya está creada. El docente ya está matriculado en el curso.

Postcondiciones: el sistema mostrará el bloque de la U.A. la tarea creada.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón "TAREA" dentro del área de la U.A..	2	Despliega una pantalla para generar la nueva tarea con campos a completar (nombre y consigna).
3	El Docente completa los campos y hace clic en botón "ENVIAR".	4	Guarda la configuración, y muestra la pantalla del curso el link de la tarea creada.

Tabla B.23: Crear una Tarea.

Apéndice B. Análisis de casos de uso por áreas

B.4.2. Modificar una Tarea.

Una vez creada la tarea es posible modificar alguno de sus campos.

Descripción completa

Caso de uso: ModificarUnaTarea

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** modifica la tarea seleccionada.
- **SISTEMA:** guarda en la base de datos las modificaciones correspondientes.

Precondiciones: la tarea que se desea modificar fue previamente creada.

Postcondiciones: se guardan las modificaciones realizadas y se muestran en pantalla.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en la tarea que desea modificar.	2	Muestra en la pantalla el formulario de la tarea seleccionada.
3	El Docente realiza las modificaciones y hace clic en el botón “ACTUALIZAR”.	4	Guarda los datos modificados de la tarea en su base de datos sobrescribiendo los anteriores.
		5	Despliega un mensaje indicando que los datos fueron actualizados.

Tabla B.24: Modificar una Tarea.

B.4.3. Hacer visible/invisible una tarea.

Una vez creada la tarea es posible modificar la visibilidad de la misma.

Descripción completa

Caso de uso: HacerVisibleInvisibleUnaTarea

Actor principal: Docente

Personal involucrado e intereses:

- DOCENTE: hace visible la tarea seleccionada.
- SISTEMA: se guardan las modificaciones realizadas.

Precondiciones: la tarea que se desea hacer visible fue previamente creada.

Postcondiciones: se guardan las modificaciones realizadas y se muestran en pantalla el link de la tarea con un ícono indicativo del estado.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en la tarea que desea hacer visible.	2	Muestra en la pantalla el formulario de la tarea seleccionada.
3	El Docente hace clic en el botón “MOSTRAR/OCULTAR”.	4	El sistema guarda en la base de datos la visibilidad de la tarea y muestra visible la tarea correspondiente en la pantalla principal del curso para alumnos y el link con el ícono indicativo del estado a los docentes.

Flujo alternativo:

3a	<p>El Docente se arrepiente hacer visible la tarea.</p> <p>3a1: El Docente hace clic nuevamente en el botón “MOSTRAR/OCULTAR”.</p>	3a2:	<p>El sistema guarda en la base de datos la no visibilidad de la tarea y no muestra la tarea correspondiente en la pantalla principal del curso para alumnos. Muestra el link con el ícono indicativo del estado a los docentes.</p>
-----------	--	------	--

Tabla B.25: Hacer visible/invisible una tarea.

B.5. Manejo del texto de referencia

En cada curso hay unidades de aprendizaje en las que se crean los textos de referencia, estos textos están asociados a una tarea dada por el docente y contienen todos los conceptos e información correspondiente a esa tarea asignada.

Alumnos y docentes crean de forma colaborativa estos textos en sublenguaje mientras la tarea esté activa. Una vez que el texto es finalizado, que el docente entiende que es un texto completo, puede ocultar la tarea impidiendo que los estudiantes sigan realizando modificaciones en los mismos. Estos textos serán luego la referencia en la instancia de evaluación.

B.5.1. Editar texto de referencia.

Los textos de referencia pueden ser modificados por alumnos y docentes. El acceso a la edición del texto es regulado por el botón “LIBRE/OCUPADO” que funciona de la siguiente manera: el botón puede tener dos colores, verde ó rojo.

Cuando está verde, cualquier usuario puede comenzar a editarlo. Una vez que comienza la edición, se preserva el color únicamente para él y los restantes usuarios ven el botón en rojo.

El botón rojo indica que el texto está siendo editado por alguien y no se permite la edición por parte de otro usuario hasta ver el botón en verde nuevamente.

Descripción completa

Caso de uso: EditarTextoDeReferencia

Actor principal: Usuario

Personal involucrado e intereses:

- USUARIO: edita el texto de referencia
- SISTEMA: guarda las modificaciones del texto de referencia de la tarea correspondiente, sobrescribiendo la versión anterior.

Precondiciones: el usuario ya está registrado en el sistema KLEARapp. Está activa la tarea para permitir editar el texto de referencia.

Postcondiciones: queda guardado el texto de referencia editado y todos lo pueden ver.

B.5. Manejo del texto de referencia

Flujo básico:

ACTOR		SISTEMA	
1	El Usuario hace clic en el ícono de la tarea a la cual le quiere escribir el texto de referencia.	2	Muestra en pantalla el formulario de la tarea correspondiente con el botón de “LIBRE/OCUPADO” en verde.
3	El Usuario hace clic en el área de edición y comienza a escribir.	4	Muestra el botón de “LIBRE/OCUPADO” en verde al Usuario que actualmente está editando y en rojo al resto de los usuarios.
5	El Usuario al terminar hace clic en “GUARDAR”.	6	Guarda el texto de referencia en la tarea correspondiente. Refresca fecha y hora de la última actualización. Muestra el botón de “LIBRE/OCUPADO” en verde para los demás usuarios.

Flujo alternativo:

	2a2:El Usuario hace clic en el área de edición.	2a	Existe un usuario que está en ese momento editando el texto. 2a1:Muestra en pantalla el formulario de la tarea correspondiente con el botón de “LIBRE/OCUPADO” en rojo. 2a3: Despliega un mensaje indicando que el texto no puede editarse hasta que el botón “LIBRE/OCUPADO” esté en verde.
5a	El Usuario que está editando el texto deja inactiva la aplicación. 5a1: El Usuario no guarda los cambios realizados y deja la aplicación sin actividad por un tiempo. El sistema guarda en su base de datos las modificaciones realizadas y retorna a la pantalla del curso.		5a2: Luego de cierto tiempo de espera pasa el botón a verde para todos los usuarios. Y le despliega al usuario que se encontraba editando el texto de referencia un mensaje indicando que se liberó la edición para todos.

Tabla B.26: Editar texto de referencia.

Apéndice B. Análisis de casos de uso por áreas

B.5.2. Descongelar texto de referencia.

El docente propone una tarea y permite que el texto de referencia se comience a crear con la participación activa de los alumnos.

Descripción completa

Caso de uso: DescongelarTextoDeReferencia

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** Descongela el texto de referencia.
- **SISTEMA:** Muestra el texto de referencia en formato de edición.

Precondiciones: el docente creó la tarea y ella se encuentra congelada.

Postcondiciones: alumnos pueden ver la tarea activa, lista para editar texto.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón de “CONGELAR/DESCONGELAR” sin haberse ponderado el grafo asociado al texto de referencia.	2	Muestra el botón de “LIBRE/OCUPADO” en verde a todos los usuarios y muestra el texto de referencia en formato de edición.

Flujo alterntivo:

1a	<p>Estando ponderado el grafo asociado al texto de referencia</p> <p>1a1: El Docente hace clic en el botón de “CONGELAR/DESCONGELAR”.</p> <p>1a3: El Docente hace clic en “SÍ”.</p> <p>1a3a: El Docente no desea borrar el grafo ponderado.</p> <p>1a3a1: El Docente hace clic en “NO”.</p>	<p>1a2: Despliega un mensaje avisando que se borrará el grafo ponderado y preguntándole si desea continuar con la acción.</p> <p>1a4: Cierra el mensaje, elimina el grafo ponderado y descongela el texto referencia.</p> <p>1a3a2: Cierra el mensaje y muestra la pantalla de la tarea sin realizar modificaciones (no descongeló el texto).</p>
-----------	---	---

Tabla B.27: Descongelar texto de referencia.

B.5.3. Congelar texto de referencia.

Una vez que el Docente entiende que el texto de referencia está completo, puede congelar la tarea impidiendo que los estudiantes sigan realizando modificaciones en los mismos. Una vez en este estado y una vez generado el grafo correspondiente, el docente tiene la posibilidad de asignar los pesos a cada nodo y arco.

Descripción completa

Caso de uso: CongelarextoDeReferencia

Actor principal: Docente

Personal involucrado e intereses:

- DOCENTE: Congela el texto de referencia que se encuentra descongelado.

Apéndice B. Análisis de casos de uso por áreas

- SISTEMA: Muestra el texto de referencia sin posibilidad de edición.

Precondiciones: el texto de referencia ha sido escrito en forma colaborativa entre alumnos y docente.

El docente se encuentra dentro de la tarea y considera finalizado el texto de referencia.

Postcondiciones: queda guardado el texto de referencia y se muestra en pantalla el link de la tarea con un ícono indicativo del estado.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón de “CONGELAR/DESCONGELAR” sin haberse ponderado el grafo asociado al texto de referencia.	2	Muestra el botón de “LIBRE/OCUPADO” en la versión congelada y muestra el texto de referencia sin posibilidad de editarlo.

Flujo alternativo:

1a	<p>Estando ponderado el grafo asociado al texto de referencia</p> <p>1a1: El Docente hace clic en el botón de “CONGELAR/DESCONGELAR”.</p> <p>1a3: El Docente hace clic en “SÍ”.</p> <p>1a3a: El Docente no desea borrar el grafo ponderado.</p> <p>1a3a1: El Docente hace clic en “NO”.</p>		<p>1a2: Despliega un mensaje avisando que se borrará el grafo ponderado y preguntándole si desea continuar con la acción.</p> <p>1a4: Cierra el mensaje, elimina el grafo ponderado y descongela el texto referencia.</p> <p>1a3a2: Cierra el mensaje y muestra la pantalla de la tarea sin realizar modificaciones (no descongeló el texto).</p>
-----------	---	--	---

Tabla B.28: Congelar texto de referencia.

B.5.4. Generar grafo del texto.

Los usuarios pueden solicitar generar y ver grafo asociado a los textos que escriben, ya sea en el texto de referencia o en sus borradores particulares. Éstos son una representación gráfica de los conceptos contenidos en los textos

Descripción completa

Caso de uso: GenerarGrafoDelTexto

Actor principal: Usuario

Personal involucrado e intereses:

- USUARIO: Solicita generar y ver grafo de un texto.
- SISTEMA: Genera y muestra el grafo del texto correspondiente.

Precondiciones: el texto ha sido escrito y el usuario se encuentra visualizándolo.

Postcondiciones: el usuario puede ver el grafo como imagen.

Flujo básico:

ACTOR		SISTEMA	
1	El Usuario hace clic en el botón “GUARDAR Y MOSTRAR GRAFO” que se encuentra junto al texto.	2	Muestra en pantalla el grafo asociado al texto.

Tabla B.29: Generar grafo del texto.

B.5.5. Asignar pesos a nodos y arcos.

Una vez que el texto de referencia es finalizado el docente genera el grafo y asigna pesos a nodos y arcos. Es decir, le asigna un valor numérico a las partes que componen el grafo, y con esto se realizará la comparación en instancia de evaluación para obtener la calificación.

Descripción completa

Caso de uso: AsignarPesosANodosYArcos

Apéndice B. Análisis de casos de uso por áreas

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** Asigna un valor numérico a cada elemento del grafo asociado al texto de referencia.
- **SISTEMA:** Guarda los datos ingresados por el docente.

Precondiciones: el texto de referencia ha sido creado y se encuentra congelado.

El grafo asociado al texto ha sido generado y el docente está en esa pantalla.

Postcondiciones: queda guardado en el sistema el grafo ponderado asociado al texto de referencia.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón de “ASIGNAR PESOS” de la pantalla del grafo asociado al texto.	2	Da la posibilidad de realizar una asignación de pesos primaria por categoría léxica de los nodos asociados al grafo.
3	El Docente introduce los valores y hace clic en el botón “LISTO!”.	4	Guarda en base de datos el grafo ponderado asociado al texto de referencia.

Tabla B.30: Asignar pesos a nodos y arcos.

B.5.6. Asignar pesos a nodos de forma particular.

El docente puede realizar una asignación seleccionando el nodo y asignándole el peso que considere conveniente según la importancia del concepto.

Descripción completa

Caso de uso: AsignarPesosANodosDeFormaParticular

Actor principal: Docente

Personal involucrado e intereses:

B.5. Manejo del texto de referencia

- **DOCENTE:** Asigna un valor numérico a un elemento específico del grafo asociado al texto de referencia.
- **SISTEMA:** Guarda los datos ingresados por el docente.

Precondiciones: el texto no ha sido descongelado luego de la última asignación. El docente se encuentra en la pantalla de asignación de pesos.

Postcondiciones: queda guardado en el sistema el grafo nuevamente ponderado asociado al texto de referencia reescribiendo la última versión.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente escribe en el buscador el nodo que desea asignarle el peso particular	2	Si la palabra ingresada forma parte de las que integran el texto de referencia, muestra que categoría léxica tiene y cual es su peso asignado de forma primaria.
3	El Docente introduce el nuevo valor y hace clic en el botón "PESO".	4	Guarda en base de datos el grafo ponderado asociado al texto de referencia.

Tabla B.31: Asignar pesos a nodos de forma particular.

B.5.7. Modificar pesos a nodos y arcos.

El docente puede asignar nuevos pesos a nodos y arcos manteniendo congelado el texto de referencia.

Descripción completa

Caso de uso: ModificarPesosANodosYArcos

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** Asigna nuevo valor numérico a elementos del grafo asociado al texto de referencia.
- **SISTEMA:** Guarda los datos ingresados por el docente.

Apéndice B. Análisis de casos de uso por áreas

Precondiciones: el docente se encuentra en la pantalla de consulta de asignación de pesos.

Postcondiciones: queda guardado en el sistema el grafo nuevamente ponderado asociado al texto de referencia reescribiendo la última versión.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón “MODIFICAR PESOS”	2	El sistema muestra la pantalla de asignación de pesos.
3	El Docente introduce los nuevos valores y hace clic en “LISTO”.	4	Guarda en base de datos los nuevos valores.

Tabla B.32: Modificar pesos a nodos y arcos.

B.6. Textos borradores

El objetivo de esta herramienta es dar a los participantes un espacio de práctica. Alumnos y docentes generarán textos y grafos relacionados con una tarea específica. Los borradores pueden servir para ser usados en los textos referencia.

B.6.1. Crear texto borrador.

El usuario puede crear un texto borrador.

Descripción completa

Caso de uso: CrearTextoBorrador

Actor principal: Usuario

Personal involucrado e intereses:

- USUARIO: crea el texto borrador.
- SISTEMA: guarda el texto borrador en el directorio BORRADORES del usuario.

Precondiciones: los alumnos y docentes se encuentran matriculados.

Postcondiciones: el sistema guarda los borradores asociados al usuario.

Flujo básico:

ACTOR		SISTEMA	
1	El Usuario hace clic en el botón “BORRADOR” que se encuentra dentro de cada U.A.	2	Muestra la pantalla que contiene el área de edición para realizar el borrador.
3	El Usuario escribe el borrador y cuando finaliza hace clic en “GUARDAR”.	4	Guarda el borrador asociado a la U.A. en la base de datos de borradores.

Tabla B.33: Crear texto borrador.

Apéndice B. Análisis de casos de uso por áreas

B.6.2. Modificar texto borrador.

Una vez que se ha creado un borrador, el usuario tiene la posibilidad de editar el mismo.

Descripción completa

Caso de uso: ModificarTextoBorrador

Actor principal: Usuario

Personal involucrado e intereses:

- USUARIO: modifica el texto borrador.
- SISTEMA: guarda los cambios realizados en el borrador.

Precondiciones: el usuario ya ha creado un borrador.

Postcondiciones: se guardaron las modificaciones del borrador y se muestran en pantalla.

Flujo básico:

ACTOR		SISTEMA	
1	El Usuario hace clic en el botón “BORRADOR” que se encuentra dentro de cada U.A.	2	Muestra la pantalla que contiene el área de edición para realizar el borrador.
3	El Usuario escribe el borrador y cuando finaliza hace clic en “GUARDAR”.	4	Guarda el borrador asociado a la U.A. en la base de datos de borradores.

Tabla B.34: Modificar texto borrador.

B.7. Léxico y reglas.

Cada curso tiene asociado una lista de léxicos.

En esta lista se basa la corrección que realiza el editor de texto en las instancias de redacción de borradores, textos de referencia y evaluaciones.

Puede realizarse una consulta de alguna palabra en particular para conocer su categoría léxica y agregar palabras si éstas no estuvieran en la lista.

B.7.1. Administrar léxicos en la aplicación.

Se da la posibilidad de modificar la lista, ya sea agregando nuevas palabras o cambiando su categoría léxica.

Descripción completa

Caso de uso: AdministrarLéxicosEnLaAplicación

Actor principal: Usuario

Personal involucrado e intereses:

- USUARIO: modifica o consulta la lista de léxico.
- SISTEMA: guarda las modificaciones en la lista de léxicos.

Precondiciones: existe una lista de léxicos asociada al curso.

Postcondiciones: cambia en la base de datos la lista de léxicos.

Flujo básico:

ACTOR		SISTEMA	
1	El usuario hace clic en el botón de “LÉXICOS Y REGLAS” del menú lateral.	2	Muestra en pantalla un buscador para ingresar la palabra.
3	Ingresa la palabra en el campo asignado para ello y hace clic en el ícono del buscador.	4	El sistema da la posibilidad de agregarla o cambiar su categoría léxica.
5	El usuario realiza la acción deseada.	6	El sistema responde de acuerdo a la opción elegida.

Tabla B.35: Administrar léxicos en la aplicación.

Apéndice B. Análisis de casos de uso por áreas

B.7.2. Corregir léxico y sintaxis en un texto (texto de referencia o borrador).

A medida que el usuario edita un texto el sistema realiza la validación del mismo. Va corrigiendo el léxico utilizado y la sintaxis, en caso de que ocurra un error el mismo despliega una lista con opciones para poder corregir.

El docente también tiene la posibilidad de agregar nuevas palabras a la lista de léxicos del sistema.

Descripción completa

Caso de uso: CorregirLéxicoEnUnTexto

Actor principal: Usuario

Personal involucrado e intereses:

- USUARIO: visualiza las observaciones sobre el texto ingresado.
- SISTEMA: detecta palabras incorrectas y brinda acciones frente a esta palabra.

Precondiciones: existe una lista de léxicos asociada al curso.
Existe un texto ingresado por el usuario.

Postcondiciones: el texto ingresado no contiene errores de léxico.

Flujo básico:

ACTOR		SISTEMA	
1	El usuario introduce texto.	2	El sistema muestra las palabras que no están ingresadas en la base de datos del léxico o las frases parciales.
3	El usuario corrige el texto o ingresa la palabra en la lista de léxicos y se le despliega un menú con las opciones a manejar.	4	El sistema da la posibilidad de agregarla.

Tabla B.36: Corregir léxico y sintaxis en un texto.

B.8. Administración de evaluación

Cada U.A., tiene asociada una evaluación con el fin de poner a prueba a los alumnos sobre el conocimiento adquirido durante el desarrollo de la tarea.

El texto de referencia llevado a cabo entre los estudiantes y el docente es el material en el cual se basa en forma exclusiva esta evaluación.

B.8.1. Crear evaluación.

El docente crea una evaluación dentro del bloque correspondiente a una U.A.

Descripción completa

Caso de uso: CrearUnaEvaluación

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** crea una Evaluación.
- **SISTEMA:** genera y muestra el link de la evaluación dentro de la U.A correspondiente.

Precondiciones: el texto de referencia ya está finalizado. El docente ya está matriculado en el curso.

Postcondiciones: el sistema mostrará en el bloque de la U.A. la evaluación creada.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón “EVALUACION” dentro del área de la U.A..	2	Despliega una pantalla para generar la nueva evaluación con campos a completar (nombre y preguntas).
3	El Docente completa los campos y hace clic en botón “CONTINUAR”.	4	Guarda la configuración, y muestra en la pantalla del curso el link de la evaluación creada.

Tabla B.37: Crear evaluación.

Apéndice B. Análisis de casos de uso por áreas

B.8.2. Modificar evaluación.

El usuario en rol de docente modifica una evaluación previamente creada.

Descripción completa

Caso de uso: ModificarEvaluacion

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** edita el formulario de la evaluación.
- **SISTEMA:** guarda las modificaciones realizadas a la evaluación.

Precondiciones: existe la evaluación que se desea modificar.

El docente ya está matriculado en el curso y se encuentra en la pantalla del curso en el que va a modificar una evaluación.

Postcondiciones: la evaluación ha sido modificada exitosamente, reflejándose esto en el cambio de algún campo del formulario de la misma.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el link de la evaluación que desea modificar.	2	Muestra en la pantalla el formulario de la evaluación seleccionada.
3	El Docente hace clic en el botón “EDITAR EVALUACION”.	4	El sistema muestra en pantalla el formulario de evaluación para realizar las modificaciones.
6	El Docente realiza los cambios y hace clic en el botón “ACTUALIZAR”	7	Guarda los datos modificados de la evaluación en su base de datos sobrescribiendo los anteriores y muestra la pantalla con la evaluación.

Flujo alternativo:

6a	El Docente no quiere realizar ninguna modificación. 1a1: El Docente hace clic en el botón “VOLVER A LAS PREGUNTAS”.	1a2: Muestra la pantalla con la evaluación.
-----------	--	---

Tabla B.38: Modificar evaluación.

B.8.3. Hacer visible/invisible una evaluación.

Una vez creada la evaluación es posible modificar la visibilidad de la misma.

Descripción completa

Caso de uso: HacerVisibleInvisibleUnaEvaluación

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** hace visible la evaluación seleccionada.
- **SISTEMA:** se guardan las modificaciones realizadas.

Precondiciones: la evaluación que se desea hacer visible fue previamente creada.

Postcondiciones: se guardan las modificaciones realizadas y se muestra en pantalla el link de la evaluación con un ícono indicativo del estado.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en la evaluación que desea hacer visible.	2	Muestra en la pantalla el formulario de la evaluación seleccionada.
3	El Docente hace clic en el botón “MOSTRAR/OCULTAR”.	4	El sistema guarda en la base de datos la visibilidad de la evaluación y muestra visible la evaluación correspondiente en la pantalla principal del curso para alumnos y el link con el ícono indicativo del estado a los docentes.

Flujo alternativo:

3a	<p>El Docente se arrepiente hacer visible la evaluación.</p> <p>3a1: El Docente hace clic nuevamente en el botón “MOSTRAR/OCULTAR”.</p>	<p>3a2: El sistema guarda en la base de datos la no visibilidad de la evaluación y no muestra la evaluación correspondiente en la pantalla principal del curso para alumnos. Muestra el link con el ícono indicativo del estado a los docentes.</p>
-----------	---	---

Tabla B.39: Hacer visible/invisible una evaluación.

B.8.4. Realizar la evaluación.

El alumno realiza la evaluación propuesta por el docente.

Descripción completa

Caso de uso: RealizarLaEvaluación

Actor principal: Alumno

Personal involucrado e intereses:

- ALUMNO: realiza la evaluación propuesta.
- SISTEMA: guarda las respuestas dadas a la evaluación.

Precondiciones: la evaluación fue previamente creada y es el momento de realizarla.

Postcondiciones: se guardan la respuestas realizadas.

Flujo básico:

ACTOR		SISTEMA	
1	El Alumno hace clic en el link de la evaluación que debe realizar.	2	Muestra en la pantalla el formulario de la evaluación con las preguntas.
3	El Alumno hace clic en la pregunta para responder.	4	Muestra en pantalla el formulario para contestar.
5	El Alumno hace clic en el área de edición y comienza a escribir, luego hace clic en el botón “VALIDAR” y “VOLVER A LAS PREGUNTAS”.	6	Muestra en la pantalla el formulario de la evaluación con las preguntas para seguir contestando.
7	El Alumno finaliza la evaluación y hace clic en “ENVIAR”.	8	Guarda las respuestas en la base de datos y despliega un mensaje indicando que la evaluación fue enviada.

Tabla B.40: Realizar la evaluación.

Apéndice B. Análisis de casos de uso por áreas

B.8.5. Docente accede a las calificaciones de la evaluación asociada a una U.A.

Una vez finalizado el período de evaluación de una determinada U.A., el docente accede a un listado con las calificaciones de los alumnos.

Descripción completa

Caso de uso: DocenteAccedeALasCalificacionesDeLaEvaluaciónAsociadaAUnaUA

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** accede a una lista con el nombre de los alumnos del curso con la nota que obtuvieron en la evaluación.
- **SISTEMA:** muestra en pantalla la lista de los alumnos con la nota que obtuvieron en la evaluación.

Precondiciones: el docente de un curso ha hecho clic en el vínculo “EVALUACIÓN” dentro de la U.A. que está siendo evaluada

Postcondiciones: se muestra en pantalla una lista de los alumnos del curso con la calificación que obtuvieron en la evaluación.

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic en el botón “CALIFICACIONES” dentro de la pantalla de la evaluación de la U.A.	2	Despliega una pantalla con las calificaciones que obtuvieron los alumnos del curso en dicha evaluación.

Tabla B.41: Docente accede a las calificaciones de la evaluación.

B.8.6. Docente accede a la evaluación de un alumno.

Al docente le interesa no sólo conocer la nota de los alumnos en una evaluación, sino también en qué se equivocaron y en qué no, para saber qué puntos de la temática debe rever para un correcto aprendizaje de los alumnos. Para ello, tendrá la posibilidad de acceder a lo que respondió cada alumno y cuál fue el grafo

asociado a cada una de estas respuestas.

Descripción completa

Caso de uso: VerEvaluaciónDeAlumno

Actor principal: Docente

Personal involucrado e intereses:

- **DOCENTE:** accede a las respuestas de un alumno y los grafos asociados a ellas.
- **SISTEMA:** muestra las respuestas de un alumno y los grafos asociados a ellas.

Precondiciones: el docente ha ingresado al curso de interés, dentro de él a la U.A. cuya evaluación desea ver. Ha ingresado a la pantalla de evaluación y a través de ella a la lista de calificaciones por alumno.

Postcondiciones: se muestra en pantalla las respuestas de un alumno a la evaluación de una U.A..

Flujo básico:

ACTOR		SISTEMA	
1	El Docente hace clic sobre el nombre del alumno dentro de la pantalla Calificaciones para acceder a su evaluación.	2	Crea y muestra un archivo .pdf con la evaluación de ese alumno y el grafo asociado.

Tabla B.42: Docente accede a la evaluación de un alumno.

Apéndice B. Análisis de casos de uso por áreas

B.8.7. Alumno accede a la calificación que obtuvo en una evaluación.

El alumno puede consultar la calificación que obtuvo en la evaluación de una U.A.

Descripción completa

Caso de uso: ConsultarCalificacionObtenidaEnEvaluacion

Actor principal: Alumno

Personal involucrado e intereses:

- ALUMNO: consulta nota obtenida en la evaluación de una U.A.
- SISTEMA: muestra en pantalla la nota que obtuvo el alumno en la evaluación de una U.A.

Precondiciones: el alumno desea conocer la calificación obtenida en la evaluación de una U.A. y ha hecho clic en el vínculo “EVALUACIÓN” dentro de la U.A. de interés.

Postcondiciones: se muestra en pantalla la calificación obtenida por el alumno en la evaluación de una U.A.

Flujo básico:

ACTOR		SISTEMA	
1	El Alumno hace clic en el botón “CALIFICACIONES” dentro de la pantalla de la evaluación de la U.A..	2	Despliega una pantalla con la calificación que obtuvo en dicha evaluación.

Tabla B.43: Alumno accede a la calificación que obtuvo en una evaluación.

B.8.8. Alumno accede a una evaluación que realizó.

Además de la calificación obtenida, el alumno puede acceder a las respuestas que ingresó en la evaluación de una U.A. así como los grafos asociados a cada una de ellas.

Descripción completa

Caso de uso: ConsultarRespuestasIngresadasEnEvaluación

Actor principal: Alumno

Personal involucrado e intereses:

- ALUMNO: consulta las respuesta que ingresó en una evaluación.
- SISTEMA: muestra las respuestas ingresadas por el alumno en la evaluación y los grafos asociados a estas respuestas.

Precondiciones: el alumno ha ingresado a la U.A. cuya evaluación desea ver, en esta U.A. ha ingresado a la pantalla de evaluación y a través de ella a la evaluación realizada.

Postcondiciones: se muestra en pantalla las respuestas ingresadas por el alumno en la evaluación y los grafos asociados a estas respuestas.

Flujo básico:

ACTOR		SISTEMA	
1	El Alumno hace clic sobre su nombre para acceder a su evaluación dentro de la pantalla Calificaciones.	2	Despliega una pantalla con el detalle de la evaluación realizada.

Tabla B.44: Alumno accede a una evaluación que realizó.

B.9. Suministro de ayuda

El sistema cuenta con dos niveles de ayuda. El primero se dirige a evacuar dudas sobre los íconos y menús contenidos en la pantalla en la que se está trabajando. A él se accede a través del ícono “?”.

El segundo contiene ayuda sobre los distintos procedimientos que se pueden realizar en el sistema dependiendo el rol del usuario. Así, lo que pueden consultar los docentes es distinto a las consultas a las que tienen acceso los alumnos.

B.9.1. Solicitar ayuda haciendo clic en el ícono “?” de la ventana.

El usuario, ya sea alumno o docente, solicita ayuda sobre algún ícono/menú de la pantalla actual haciendo clic en el ícono “?”.

Apéndice B. Análisis de casos de uso por áreas

Descripción completa

Caso de uso: SolicitarAyudaMedianteIcono

Actor principal: Usuario

Personal involucrado e intereses:

- USUARIO: hace clic en el botón de ayuda “?”.
- SISTEMA: despliega una ventana emergente con ayuda sobre la ventana actual.

Precondiciones: el sistema KLEARapp está activo y el usuario se ha autenticado en éste.

Postcondiciones: se proporcionó ayuda sobre la pantalla en que se solicitó.

Flujo básico:

ACTOR		SISTEMA	
1	El Usuario hace clic en el ícono de ayuda “?”.	2	Despliega una ventana emergente que contiene: una descripción detallada de la funcionalidad de los íconos/menús.
3	El Usuario hace clic en “CERRAR” una vez que realiza la consulta.	4	Cierra la ventana emergente.

Tabla B.45: Solicitar ayuda desde el ícono “?”.

B.9.2. Solicitar ayuda haciendo clic en la palabra “Ayuda” del menú lateral.

El usuario accede a la ayuda a través del menú lateral y el sistema despliega una ventana emergente con ayuda general del sistema.

Descripción completa

Caso de uso: SolicitarAyudaMediantePalabraAyuda

Actor principal: Usuario

Personal involucrado e intereses:

- USUARIO: hace clic en el botón “AYUDA” de la barra lateral.
- SISTEMA: despliega una ventana emergente con ayuda general del sistema.

Precondiciones: el sistema KLEARapp está activo y el usuario se ha autenticado en éste.

Postcondiciones: se desplegó una ventana emergente que muestra un listado de ayuda general sobre el sistema.

Flujo básico:

ACTOR		SISTEMA	
1	El Usuario hace clic en el botón “AYUDA” del menú lateral.	2	Despliega una ventana emergente que contiene: Índice sobre ayuda general del sistema.
3	El Usuario hace clic sobre un ítem del índice.	4	Despliega el ítem del índice sobre el que se hizo clic.

Tabla B.46: Solicitar ayuda desde el menú lateral.

Apéndice C

Manuales de usuario

C.1. Introducción.

En este Apéndice se presenta el contenido de manual de usuario que se encuentra adjunto con la documentación.

Manual de usuario para rol “Administrador”

- Registrarse como administrador en la aplicación KLEARapp (pág.1)
- Iniciar sesión (pág.3)
- Barra de navegación (pág.4)
- Pantalla principal (pág.8)
- Menú Lateral (pág.9)

Manual de usuario para rol “Alumno”

- Registrarse como alumno en la aplicación KLEARapp (pág.1)
- Iniciar sesión (pág.3)
- Barra de navegación (pág.4)
- Matricularse en un curso (pág.8)
- Pantalla principal (pág.11)
- Menú Lateral (pág.12)
- Tareas (pág.14)
- Borradores (pág.22)
- Evaluación (pág.25)

Apéndice C. Manuales de usuario

Manual de usuario para rol “Docente”

- Registrarse como docente en la aplicación (pág.1)
- Inicio de sesión (pág.3)
- Barra de navegación (pág.4)
- Matricularse en un curso (pág.8)
- Página principal del curso (pág.10)
- Menú lateral (pág.11)
- Tareas (pág.24)
- Borradores (pág.34)
- Evaluación (pág.37)

Referencias

- [1] A. D Landeta, “Buenas prácticas de e-learning”, <http://www.buenaspracticas-elearning.com/indice-buenas-practicas-e-learning.html>, 2007.
- [2] Víctor González Barbone, “E-assessment of free text answers based on domain specific sublanguages and knowledge representation”, Ph.D. dissertation, Universidad de Vigo, Departamento de Ingeniería Telemática, 2013, [Online; accessed 2014-10-22].
- [3] Wikipedia, “Python software foundation license”, http://es.wikipedia.org/wiki/Python_Software_Foundation_License, [Online; accessed 2014-11-27].
- [4] G. R. Jesús González Barahona, Joaquín Seoane Pascual, “Introducción al software libre.” http://ocw.uoc.edu/informatica-tecnologia-y-multimedia/introduccion-al-software-libre/XP06_M2101_01492.pdf, 2003.
- [5] Raúl González Duque, “Python para todos.” [http://mundogeek.net/tutorial-python.](http://mundogeek.net/tutorial-python/), 2007.
- [6] The Natural Language Toolkit, “Nltk”, <http://www.nltk.org/>, [Online; accessed 2014-10-30].
- [7] Python community, “Python programming language -the official webside”, <http://www.python.org>, 2014, [Online; accessed 2014-10-30].
- [8] Massimo Di Pierro., “Web2py complete reference manual 6th edition.” [http://web2py.com/books/.](http://web2py.com/books/), [Online; accessed 2014-10-27].
- [9] Desarrollo de Software para Ingeniería Eléctrica, “Casos de uso,” *Desarrollo de Software*, 2009.
- [10] Eugenia Bahit, “Poo y mvc en php.” <http://eugeniabahit.blogspot.com/>, 2011, [Online; accessed 2014-10-30].
- [11] Wikipedia, “Patrones de arquitectura,” http://es.wikipedia.org/wiki/Patrones_de_arquitectura, [Online; accessed 2014-10-30].
- [12] Wikipedia, “Lógica de negocio,” http://es.wikipedia.org/wiki/Logica_de_negocio, [Online; accessed 2014-10-30].

Referencias

- [13] Miguel Angel Álvarez, “Qué es mvc”, <http://www.desarrolloweb.com/articulos/que-es-mvc.html>, enero 2014, [Online; accessed 2014-10-30].
- [14] Cake Software Foundation, “Cakephp”, <http://book.cakephp.org/>, Copyright 2014, Cake Software Foundation, Inc. Last updated on Oct 30, 2014.
- [15] Wikipedia, “Modelo-vista-controlador,” <http://es.wikipedia.org/wiki/Modelo>, [Online; accessed 2014-10-30].
- [16] M. Marqués, “Bases de datos”, <http://www.uji.es/bin/publ/edicions/bdatos.pdf>, 2011.
- [17] M. Cristiá, “Introducción al testing de software”, http://www.academia.edu/292809/Introduccion_Al_Testing_De_Software, 2009.
- [18] Microsoft, “Bases de datos”, [http://msdn.microsoft.com/en-us/library/aa292197\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/aa292197(v=vs.71).aspx), 2003.
- [19] Wikipedia, “Unit testing”, http://en.wikipedia.org/wiki/Unit_testing.
- [20] Wikipedia, “Prueba unitaria”, http://es.wikipedia.org/wiki/Prueba_unitaria.
- [21] Graphviz(2013), “Graphviz - graph visualization software.”, <http://www.graphviz.org/>
- [22] Linterna digital, “Flat Design o Diseño Plano” <http://linternadigital.com/flat-design-o-diseno-plano-la-nueva-tendencia/>, 2014.
- [23] Glyphicons, “Motivation behind icons” <http://glyphicons.com/motivation-behind-icons/>, 2010-2014.
- [24] Concept inbox , “Visual Collaboration for creative works” <http://conceptinbox.com/>
- [25] Html colores, “Códigos de colores HTML” <http://html-color-codes.info/codigos-de-colores-hexadecimales/>, 2011.

Índice de tablas

A.1. Acciones de respuesta ante los riesgos.	67
B.1. Registrarse en la aplicación con rol alumno.	70
B.2. Registrarse en la aplicación con rol docente.	72
B.3. Registrarse en la aplicación con rol administrador.	73
B.4. Solicitar matriculación a un curso.	75
B.5. Confirmar matriculación de alumno.	76
B.6. Desmatriculación de alumno de un curso.	78
B.7. Eliminación de alumno por parte de un docente.	79
B.8. Deshacer eliminación de alumno por parte de un docente.	80
B.9. Solicitar matriculación con rol docente en curso existente.	82
B.10. Confirmar matriculación de Docente.	83
B.11. Eliminar usuario docente de un curso.	85
B.12. Restablecer docente por otro docente del curso.	86
B.13. Restablecer matriculación de docente por parte del administrador.	87
B.14. Eliminación de docente por parte del administrador.	88
B.15. Crear un nuevo curso en el sistema KLEARapp.	90
B.16. Borrar un curso del sistema KLEARapp por parte de un docente.	91
B.17. Modificar curso.	92
B.18. Borrar un curso del sistema KLEARapp por parte del administrador.	93
B.19. Usuario ingresa al curso.	94
B.20. Crear una U.A.	96
B.21. Eliminar una U.A.	97
B.22. Modificar una U.A.	98
B.23. Crear una Tarea.	99
B.24. Modificar una Tarea.	100
B.25. Hacer visible/invisible una tarea.	101
B.26. Editar texto de referencia.	103
B.27. Descongelar texto de referencia.	105
B.28. Congelar texto de referencia.	106
B.29. Generar grafo del texto.	107
B.30. Asignar pesos a nodos y arcos.	108
B.31. Asignar pesos a nodos de forma particular.	109
B.32. Modificar pesos a nodos y arcos.	110
B.33. Crear texto borrador.	111

Índice de tablas

B.34.Modificar texto borrador.	112
B.35.Administrar léxicos en la aplicación.	113
B.36.Corregir léxico y sintaxis en un texto.	114
B.37.Crear evaluación.	115
B.38.Modificar evaluación.	117
B.39.Hacer visible/invisible una evaluación.	118
B.40.Realizar la evaluación.	119
B.41.Docente accede a las calificaciones de la evaluación.	120
B.42.Docente accede a la evaluación de un alumno.	121
B.43.Alumno accede a la calificación que obtuvo en una evaluación.	122
B.44.Alumno accede a una evaluación que realizó.	123
B.45.Solicitar ayuda desde el ícono "?".	124
B.46.Solicitar ayuda desde el menú lateral.	125

Índice de figuras

1.1.	Grafo del texto de referencia generado por el sistema KLEAR . . .	3
1.2.	Grafo resultante de la comparación con el texto respuesta generado por el sistema KLEAR.	4
3.1.	Identificación de Roles.	15
3.2.	Funcionalidades referentes al rol Docente.	15
3.3.	Funcionalidades referentes al rol Alumno.	16
3.4.	Funcionalidades referentes al rol Administrador.	16
4.1.	Interacción entre los componentes en el modelo MVC. Fig. extraída de [15]	22
4.2.	Diagrama de componentes del modelo MVC.	23
4.3.	Vistas asociada a las funciones contenidas en el controlador cursos.py.	24
4.4.	Vistas asociada a las funciones contenidas en el controlador textin.py.	25
4.5.	Vistas asociada a sus respectivos controladores.	26
4.6.	Funciones contenidas en el controlador cursos.py.	27
4.7.	Funciones contenidas en el controlador textin.py.	28
4.8.	Funciones contenidas en los controladores default.py y usuarios.py.	28
4.9.	Tablas correspondientes a la base de datos propuesta en el Modelo.	29
4.10.	Componentes del sistema KLEARapp.	29
4.11.	Detalle de los módulos del sistema KLEAR.	30
6.1.	Código que define los grupos en el modelo tablas.py.	36
6.2.	Vista de la pantalla principal de la aplicación.	36
6.3.	Fragmento del código que muestra cómo serializar un objeto klgraph.	38
6.4.	Tarea oculta por el docente. El ícono muestra un ojo cerrado destacado con un círculo.	39
6.5.	Tarea visible y congelada. Se destacan los íconos ojo abierto y candado.	39
6.6.	Fragmento del código que permite que un campo de un formulario no sea editable.	39
6.7.	Fragmento del código donde se importa el plugin en el controlador textin.py.	40
6.8.	Fragmento del código que muestra cómo se invoca a la función del plugin importado.	40

Índice de figuras

6.9. Pantalla donde se muestra el plugin con las sentencias para armar la respuesta.	41
6.10. El proceso general de testing. Figura extraída de [17].	42
8.1. Porcentajes de implementación de casos de uso.	51
A.1. Diagrama Gantt con el cronograma de tareas.	63
A.2. Matriz de riesgos.	66

Esta es la última página.
Compilado el martes 2 diciembre, 2014.
<http://iie.fing.edu.uy/>

KLEARapp – Manuales

KLEARapp – Manuales

Manual de usuario para rol “Administrador”

1. [Registrarse como administrador en la aplicación KLEARapp](#) (pág.1)
2. [Iniciar sesión](#) (pág.3)
3. [Barra de navegación](#) (pág.4)
 - 3.1 [Aspectos generales](#)
 - 3.2 [Cambiar perfil](#)
 - 3.3 [Cambiar contraseña](#)
 - 3.4 [Cerrar sesión](#)
4. [Pantalla principal](#) (pág. 8)
5. [Menú Lateral](#) (pág.9)

Manual de usuario para rol “Alumno”

1. [Registrarse como alumno en la aplicación KLEARapp](#) (pág.1)
2. [Iniciar sesión](#) (pág.3)
3. [Barra de navegación](#) (pág.4)
 - 3.1 [Aspectos generales](#)
 - 3.2 [Cambiar perfil](#)
 - 3.3 [Cambiar contraseña](#)
 - 3.4 [Cerrar sesión](#)
4. [Matricularse en un curso](#) (pág.8)
 - 4.1 [Para matricularse a un curso](#)
 - 4.2 [En qué estado puede encontrarse la solicitud de matriculación.](#)
 - A. [Matriculación pendiente](#)
 - B. [Matriculación activa](#)
 - C. [Matriculación borrada](#)

KLEARapp – Manuales

Manual de usuario para rol “Alumno”

5. [Pantalla principal](#) (pág.11)
6. [Menú Lateral](#) (pág.12)
7. [Tareas](#) .(pág.14)
 - 7.1 [Ingresar texto referencia](#)
 - 7.2 [Guardar grafo](#)
 - 7.3 [Abrir grafo](#)
 - 7.4 [Corrección de léxicos y reglas](#)
8. [Borradores](#) (pág.22)
 - 8.1 [Crear borrador](#)
 - 8.2 [Modificar nombre del borrador](#)
 - 8.3 [Ingresar texto al borrador](#)
9. [Evaluación](#) (pág.25)
 - 9.1 [Acceder a la evaluación](#)
 - 9.2 [Responder evaluación](#)

Manual de usuario para rol “Docente”

1. [Registrarse como docente en la aplicación](#) (pág.1)
2. [Inicio de sesión](#) (pág.3)
3. [Barra de navegación](#) (pág.4)
 - 3.1 [Cambiar perfil](#)
 - 3.2 [Cambiar contraseña](#)
 - 3.3 [Cerrar sesión](#)

KLEARapp – Manuales

Manual de usuario para rol “Docente”

4. [Matricularse en un curso](#) (pág.8)
 - 4.1 [Para matricularse a un curso](#)
 - 4.2 [En qué estado puede encontrarse la solicitud de de matriculación](#)
 - A. [Matriculación pendiente](#)
 - B. [Matriculación activa](#)
 - C. [Matriculación borrada](#)
5. [Página principal del curso](#) (pág.10)
6. [Menú lateral](#) (pág.11)
 - 6.1 [Menú participante](#)
 - 6.2 [Menú Administrativo](#)
 - 6.3 [Administrar curso](#)
 - 6.3.1 [Modificar curso](#)
 - 6.3.1.1 [Crear U.A.](#)
 - 6.3.1.2 [Modificar U.A.](#)
 - 6.3.1.3 [Eliminar U.A.](#)
 - 6.3.1.4 [Eliminar curso](#)
 - 6.3.2 [Solicitudes](#)
 - 6.3.3 [Exportación de datos](#)
 - 6.3.4 [Reactivaciones](#)
 - 6.4 [Menú General](#)

KLEARapp – Manuales

Manual de usuario para rol “Docente”

7. [Tareas](#) (pág.24)
 - 7.1 [Crear Tarea](#)
 - 7.2 [Modificar Tarea](#)
 - 7.3 [Mostrar u ocultar Tarea](#)
 - 7.4 [Congelar / Descongelar Tarea](#)
 - 7.5 [Ingresar Texto de referencia](#)
 - 7.6 [Administrar pesos](#)

8. [Borradores](#) (pág. 34)
 - 8.1 [Crear borrador](#)
 - 8.2 [Modificar nombre del borrador](#)
 - 8.3 [Ingresar texto al borrador](#)

9. [Evaluación](#) (pág. 37)
 - 9.1 [Crear Evaluación](#)
 - 9.2 [Ingresar respuesta correcta](#)
 - 9.3 [Grafo de la respuesta correcta](#)
 - 9.4 [Opciones para el grafo generado](#)

Administrador

KLEARapp

Registrarse como administrador en la aplicación KLEARapp

1. Hacer clic en “Alumno” ó “Docente”.



2. Hacer clic en el botón “Registrarse” .



KLEARapp

Registrarse como administrador en la aplicación KLEARapp (continuación)

3. Completar el formulario con sus datos. Luego hacer clic en el botón “Regístrese”.

ADVERTENCIA: Ingresar en el campo “Nombre” la palabra clave **Admin_klear2014**.

KLEARapp

Bienvenidos a KLEARapp!

KLEARapp es una aplicación web para soporte de un sistema de escritura y generación de estructuras de conocimiento

Regístrese

Nombre:

Apellido:

Correo electrónico:

Cédula de identidad:

Contraseña:

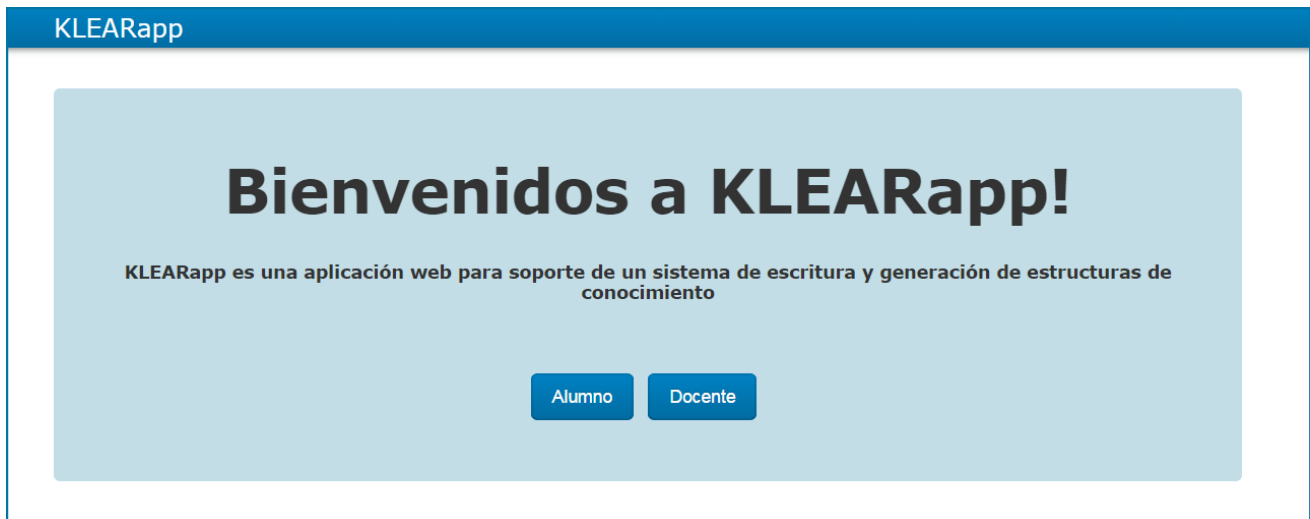
Verificar Contraseña:
 Por favor, ingrese la contraseña nuevamente

Luego se muestra la pantalla donde se listan todos los cursos que se han creado.

KLEARapp

Iniciar sesión

1. Hacer clic en alguno de los botones de la pantalla principal.



2. Hacer clic en el botón "Inicio de sesión".



KLEARapp

Barra de navegación

Aspectos generales

Una vez que el usuario ha iniciado sesión se muestra la pantalla donde se listan todos los cursos. Sobre la barra de navegación que se encuentra en la parte superior de la pantalla, en el lado derecho se puede observar el nombre del usuario que ha iniciado sesión.

KLEARapp

Bienvenido Admin_klear2014 ▾

Al posicionar el cursor sobre el nombre, se despliegan varias funcionalidades que respectan al manejo del usuario:

- [Perfil](#)
- [Contraseña](#)
- [Fin de sesión](#)

The screenshot displays the KLEARapp interface. At the top, there is a blue navigation bar with the text 'KLEARapp' on the left and 'Bienvenido Admin_klear2014 ▾' on the right. Below the navigation bar, the main content area is divided into two sections. On the left, there is a 'Menú Administrartivo' (Administrative Menu) with four items: 'Lista docentes', 'Léxicos', 'Reglas', and 'Eliminar Cursos'. On the right, there is a light blue box titled 'El Administrador:' (The Administrator:) containing a list of administrative functions: 'administra a los usuarios en rol Docente.', 'puede cambiar módulos de sublenguaje.', and 'puede eliminar cursos.'. A user menu is open over the 'Bienvenido Admin_klear2014 ▾' text, showing three options: 'Perfil', 'Contraseña', and 'Fin de sesión'.

KLEARapp

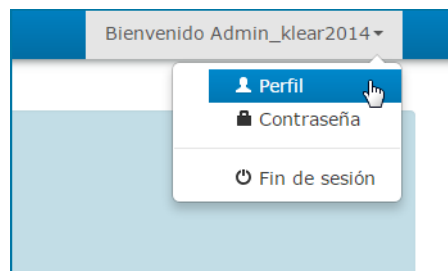
Barra de navegación (continuación)

Cambiar perfil

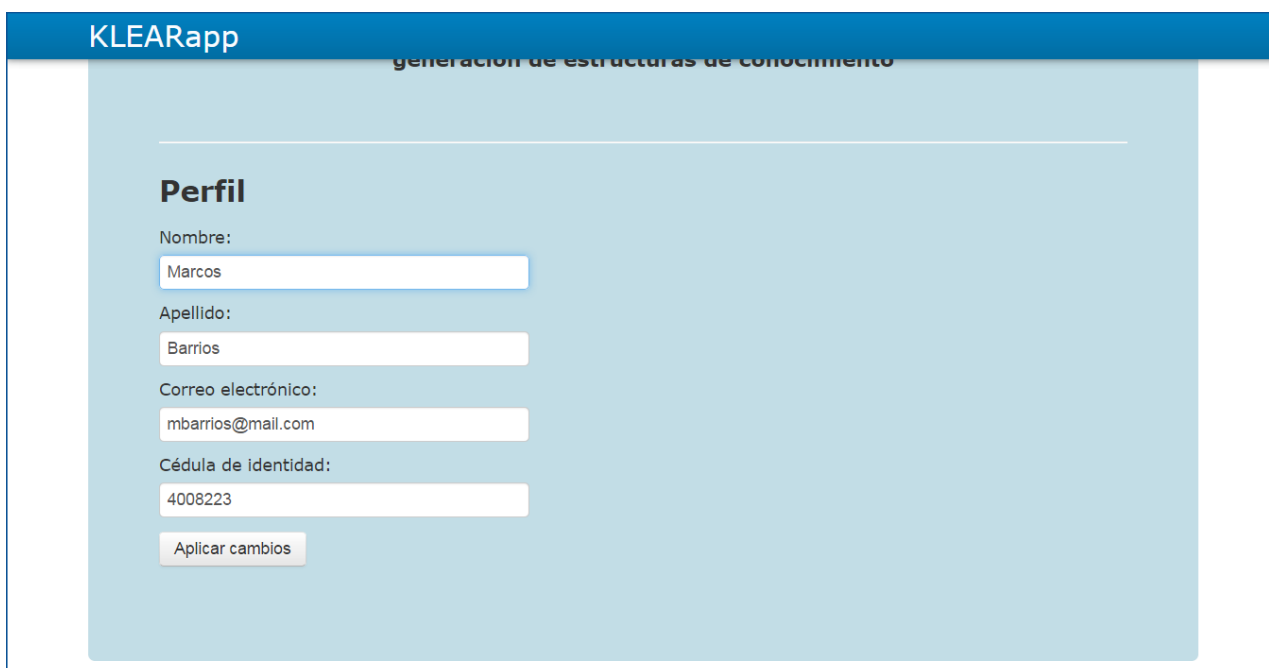
Desde la barra de navegación, al posicionarse sobre el nombre del usuario se despliega el menú con las funcionalidades relacionadas con el manejo de usuario.

Para cambiar el perfil:

1. Hacer clic en la opción **“Perfil”**.



2. El alumno puede sobrescribir aquellos campos que desea modificar.



3. Una vez hecho los cambios, hacer clic en **“Aplicar cambios”**.

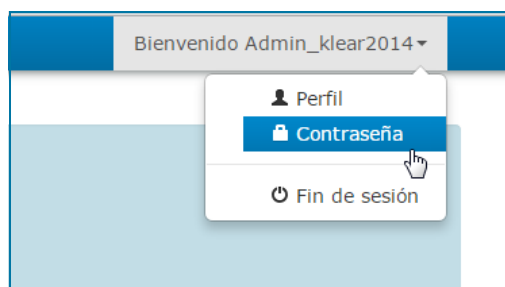
KLEARapp

Barra de navegación (continuación)

Cambiar contraseña

Para cambiar la contraseña:

1. Hacer clic en la opción **“Contraseña”**.



Complete el formulario bajo el nombre **“Cambie la contraseña”** siguiendo los pasos:

1. En el campo **“Contraseña actual”**, ingresar la contraseña con que se inició sesión.
2. En **“Nueva contraseña”** ingresar la contraseña por la que se desea cambiar.
3. En donde dice **“Verifique contraseña”**, ingresar nuevamente la nueva contraseña.
4. Para finalizar, hacer clic en el botón **“Cambiar contraseña”**

Una captura de pantalla de la interfaz de usuario de KLEARapp. En la parte superior, hay una barra azul con el texto "KLEARapp". Debajo, hay un encabezado que dice "Bienvenidos a KLEARapp!". A continuación, hay un subtítulo que dice "KLEARapp es una aplicación web para soporte de un sistema de escritura y generación de estructuras de conocimiento". El formulario principal tiene un título "Cambie la contraseña" y tres campos de entrada de texto: "Contraseña actual:", "Nueva contraseña:" y "Verificar Contraseña:". Debajo de los campos, hay un botón que dice "Cambiar contraseña".

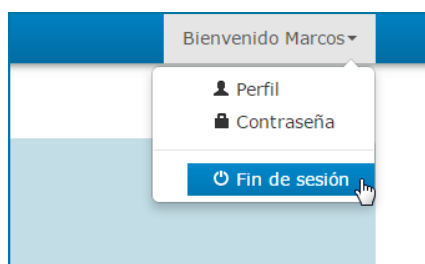
KLEARapp

Barra de navegación (continuación)

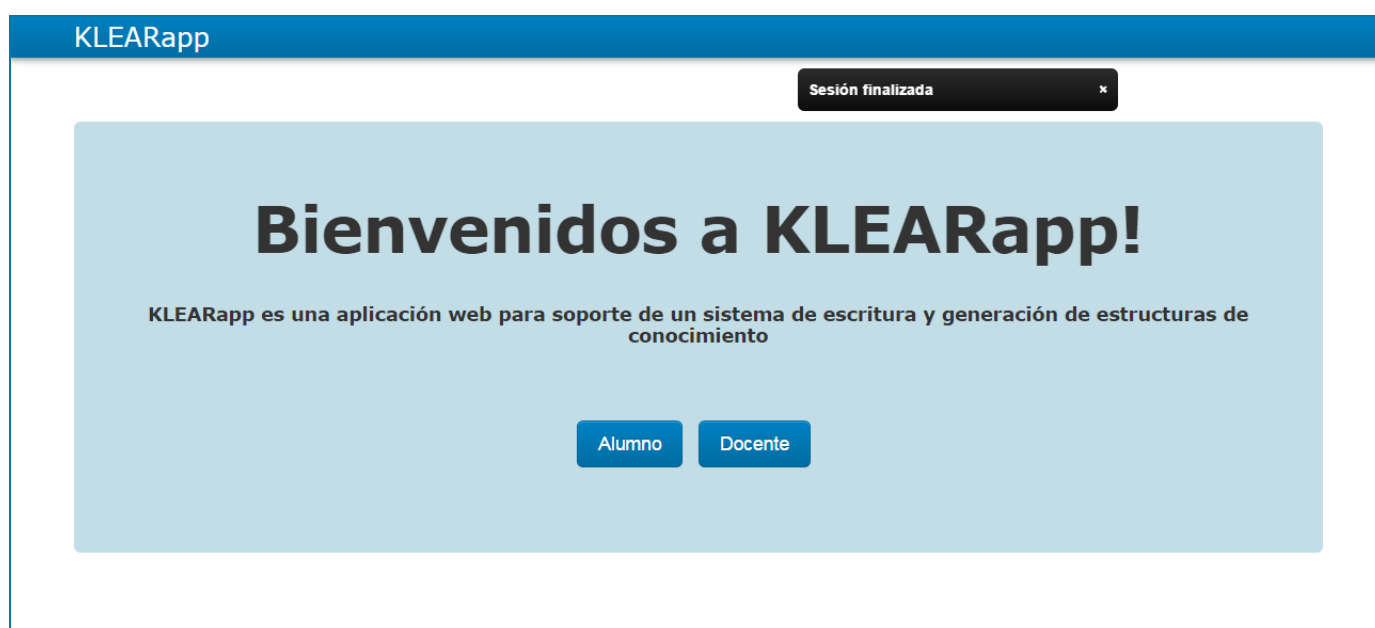
Cerrar sesión

Para cerrar sesión:

1. Hacer clic en la opción **“Fin de sesión”**.



Se redirige a la pantalla principal y se despliega en fondo negro el siguiente mensaje:

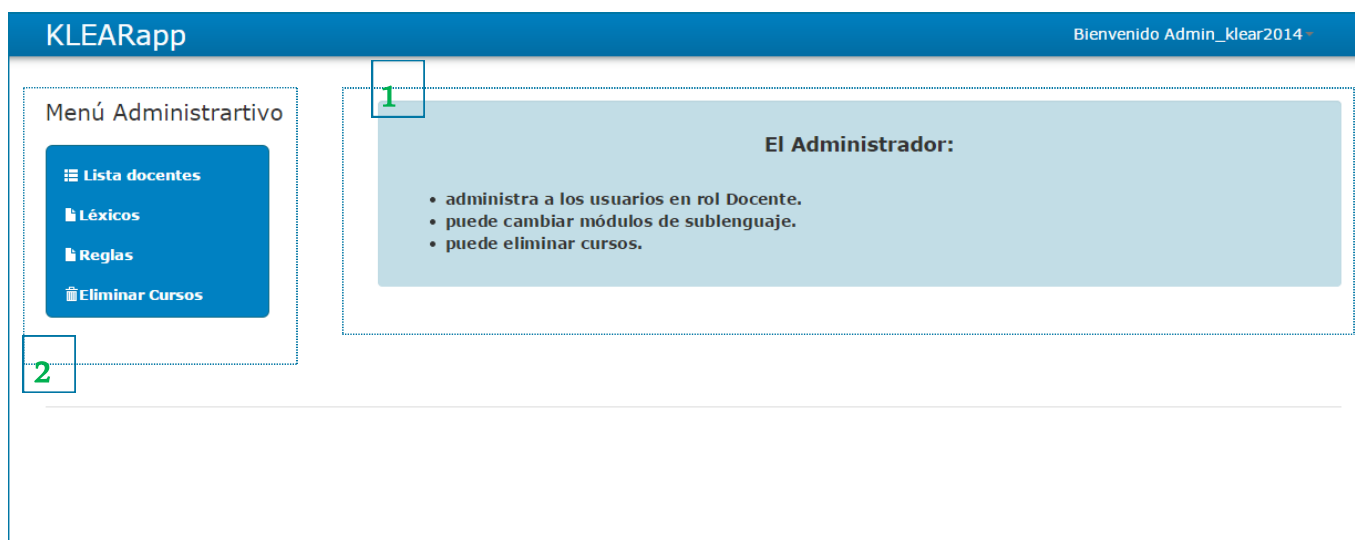


Otra forma de siempre cerrar sesión es hacer clic en el logo **“KLEARapp”**.

KLEARapp

Pantalla principal

Se muestra pantalla principal del administrador.



El administrador una vez que inicia sesión, ingresa a la pantalla principal y se encuentra con las siguientes secciones:

1. Área principal de la pantalla.
2. Menú Administrativo, que incluye:

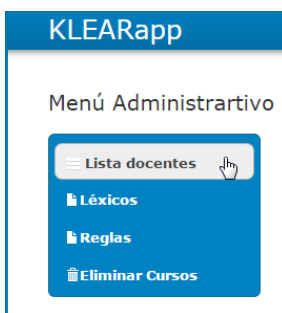
- [Lista de docentes](#)
- [Léxicos](#)
- [Reglas](#)
- [Eliminar curso](#)

KLEARapp

Menú administrativo

A través de este menú el administrador tiene acceso a varias funcionalidades.

Lista de docentes



Al hacer clic en el botón “**Lista docentes**”, en el área principal se listarán todos los docentes que haya registrados hasta el momento.

Identificador	Nombre	Estado	Curso	
1234567	Docente Docente	Activo	curso0	✎ Editar 🗑 Eliminar
1234567	Martinez, Paula	Activo	Curso0	✎ Editar 🗑 Eliminar

En esta pantalla el administrador podrá:

- **Aceptar o rechazar solicitud por parte de un usuario para ser docente en un curso.**
- **Eliminar docente de la aplicación.**

KLEARapp

Menú administrativo (continuación)

Lista de docentes (continuación)

- **Aceptar o rechazar solicitud por parte de un usuario para ser docente en un curso.**

Identificador	Nombre	Estado	Curso	
1234567	Docente Docente	Pendiente	curso0	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

1. Hacer clic en el botón **“Editar”** en la fila que corresponde a la solicitud del docente que se quiere aceptar / rechazar en el curso.
2. Se despliega una pantalla con los datos del docente seleccionado. Para modificar el estado de su matriculación, el administrador debe hacer clic en campo Estado . Se despliega un menú con las opciones de la matriculación: **“Activo”**, **“Pendiente”**, **“Borrado”**.

Lista de Docentes

Id: 1
Identificador: 1234567
Email: None
Nombre: Docente Docente
Curso: curso0
Rol: docente
Estado: ▼
Inicio:
Fin:
Marque para eliminar:

Selecciones:
Activo
Pendiente
Borrado

Estado “Activo” : Es cuando un docente tiene permiso para acceder y administrar un

Estado “Pendiente” : Es cuando un docente ha solicitado permiso para matricularse a un


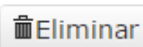
Estado “Borrado” : Es cuando se decide que el docente seleccionado no puede tener

KLEARapp

Menú administrativo (continuación)

Lista de docentes (continuación)

- Eliminar docente de la aplicación.

Identificador	Nombre	Estado	Curso	
1234567	Docente Docente	Pendiente	curso0	 

1. Hacer clic en el botón “**Eliminar**” en la fila que corresponde al docente cuyo registro se quiere borrar.

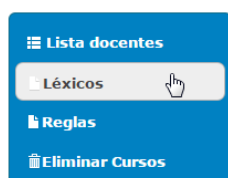
ADVERTENCIA: Una vez seleccionado el botón la acción ocurrirá de forma inmediata.

KLEARapp

Menú administrativo (continuación)

Léxicos

Menú Administrativo



Al hacer clic en el botón “**Léxico**”, se muestra en pantalla los datos del léxico que está utilizando la aplicación y se da la posibilidad de cambiarlo por otro.

Administración de archivos de Léxico

Id	Name	Lang	Descrip	Wordforms	Notes	
1	EnLong3k	English	An experimental l...	file		<input type="button" value="Vista"/> <input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

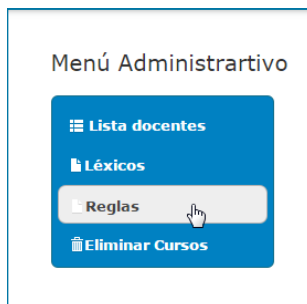
Para subir otro archivo que incluya otro léxico:

1. Hacer clic en el botón “**Editar**” .
2. Se despliega una pantalla con los datos del docente seleccionado. Para sustituir dicho archivo , se hace clic en “**Seleccionar archivo**” y se sube el léxico nuevo.
3. Para finalizar el procedimiento hacer clic en el botón “**Enviar**”.

KLEARapp

Menú administrativo (continuación)

Reglas



Al hacer clic en el botón “**Reglas**”, se muestra en pantalla los datos del archivo que contiene las reglas sintácticas que está utilizando la aplicación y se da la posibilidad de cambiarlas por otro conjunto.

Administración de archivos de Reglas

+Añadir Buscar Limpiar

Id	Name	Lang	Descrip	Rules	Examples	Notes	
1	En05Rules	English		file	file		<input type="button" value="Vista"/> <input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Para subir otro archivo que incluya otras reglas.

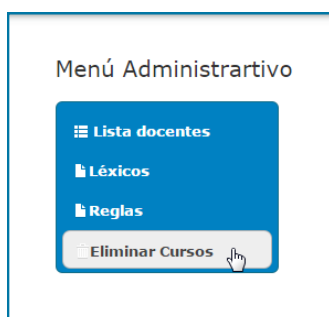
1. Hacer clic en el botón “**Editar**” .
2. Se despliega una pantalla con los datos del docente seleccionado. Para sustituir dicho archivo , se hace clic en “**Seleccionar archivo**” y se sube las nuevas reglas.
3. Para finalizar el procedimiento hacer clic en el botón “**Enviar**”.

La lista de **léxicos** y **reglas** permiten el buen funcionamiento del sublenguaje.

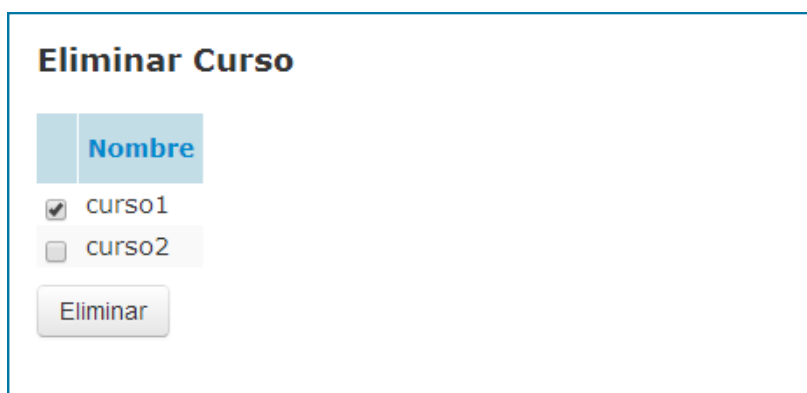
KLEARapp

Menú administrativo (continuación)

Eliminar Cursos



Al hacer clic en el botón “**Eliminar Cursos**”, se le da la posibilidad al administrador de eliminar uno o varios cursos a la vez.



Para eliminar uno o varios cursos:

1. Seleccionar aquellos cursos que se desea eliminar.
2. Hacer clic “**Eliminar**”.

ADVERTENCIA: Una vez seleccionado el botón la acción ocurrirá de forma inmediata.

Alumno

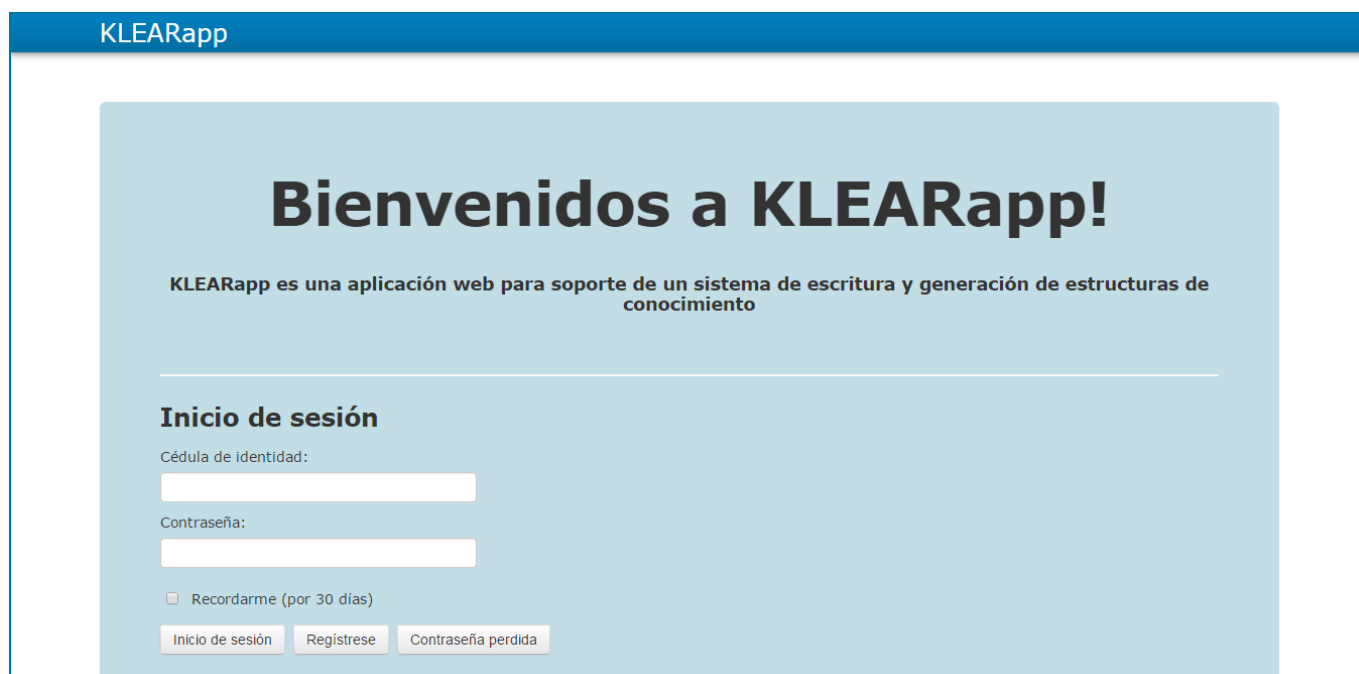
KLEARapp

Registrarse como alumno en la aplicación KLEARapp

1. Hacer clic en el botón “**Alumno**” de la pantalla principal.



2. Hacer clic en el botón “**Registrarse**” .



KLEARapp

Registrarse como alumno en la aplicación KLEARapp (continuación)

3. Completar el formulario con sus datos. Luego hacer clic en el botón “Regístrese”.



KLEARapp

Bienvenidos a KLEARapp!

KLEARapp es una aplicación web para soporte de un sistema de escritura y generación de estructuras de conocimiento

Regístrese

Nombre:

Apellido:

Correo electrónico:

Cédula de identidad:

Contraseña:

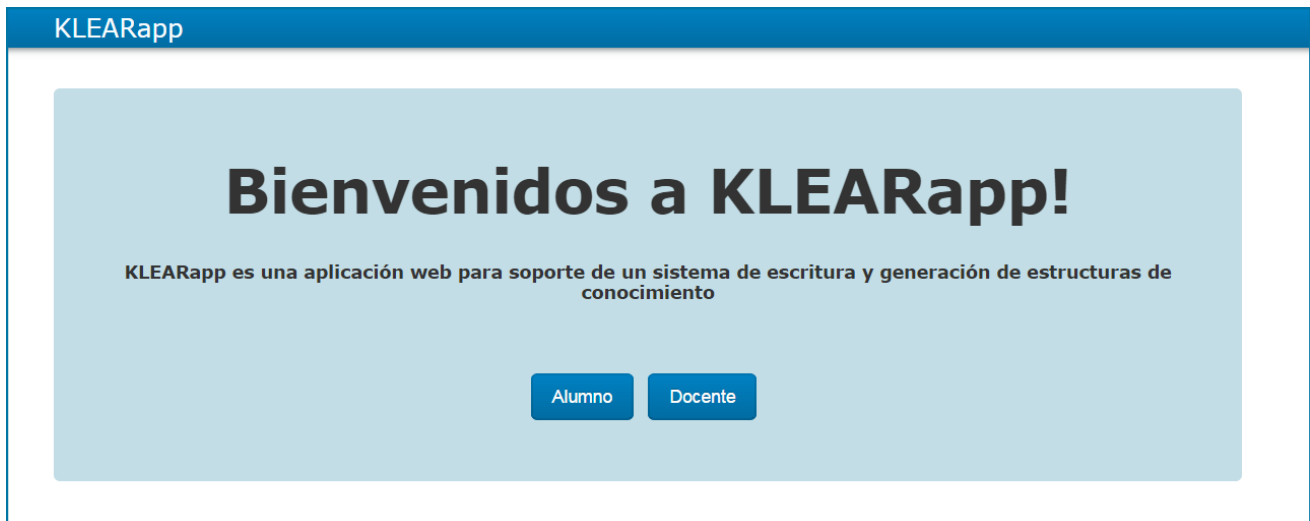
Verificar Contraseña:
 Por favor, ingrese la contraseña nuevamente

Luego se muestra la pantalla donde se listan todos los cursos que se han creado.

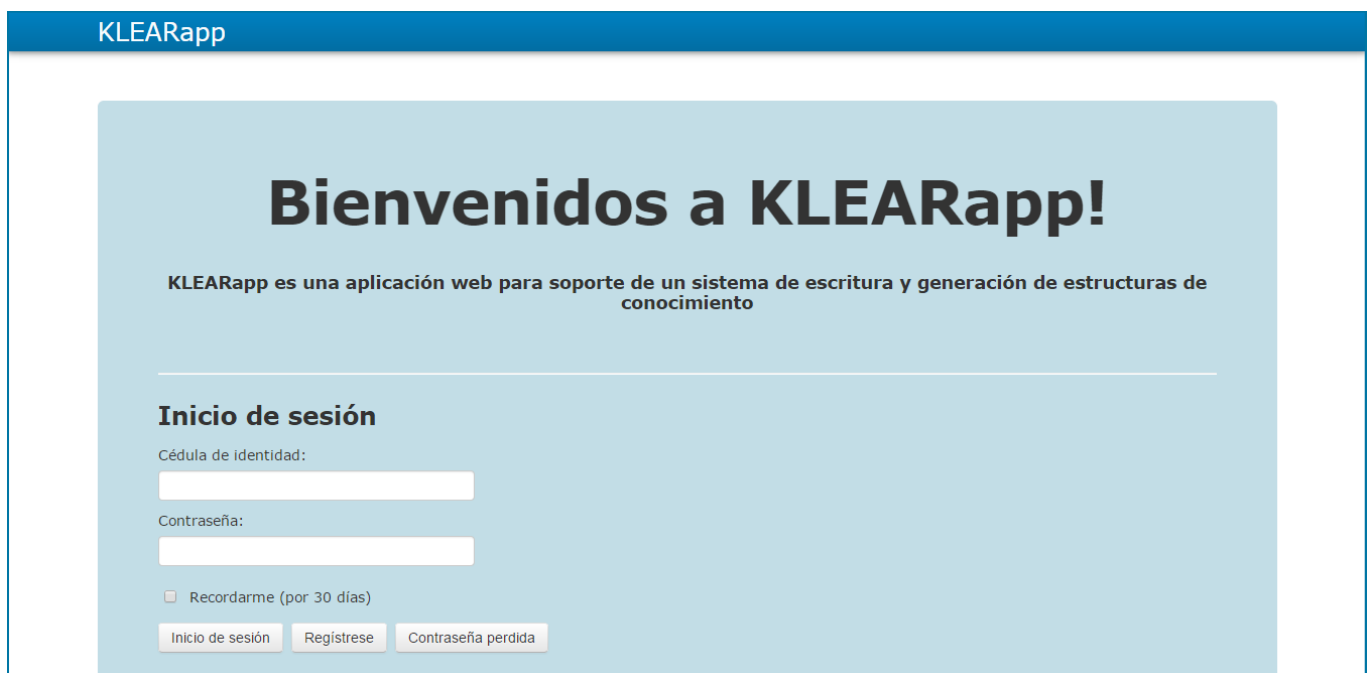
KLEARapp

Iniciar sesión

1. Hacer clic en el botón “Alumno”.



2. Hacer clic en el botón “Inicio de sesión” .



KLEARapp

Barra de navegación

Aspectos generales

Una vez que el usuario ha iniciado sesión se muestra la pantalla donde se listan todos los cursos. Sobre la barra de navegación que se encuentra en la parte superior de la pantalla en el lado derecho, se puede observar el nombre del usuario que ha iniciado sesión.

KLEARapp

Bienvenido Marcos ▾

Al posicionar el cursor sobre el nombre se despliegan varias funcionalidades que respectan al manejo del usuario:



KLEARapp

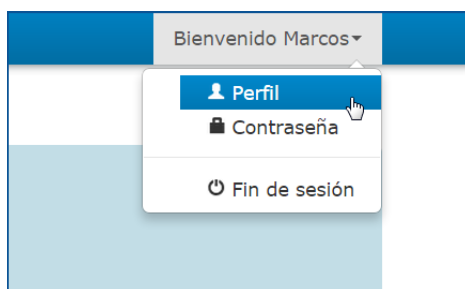
Barra de navegación (continuación)

Cambiar perfil

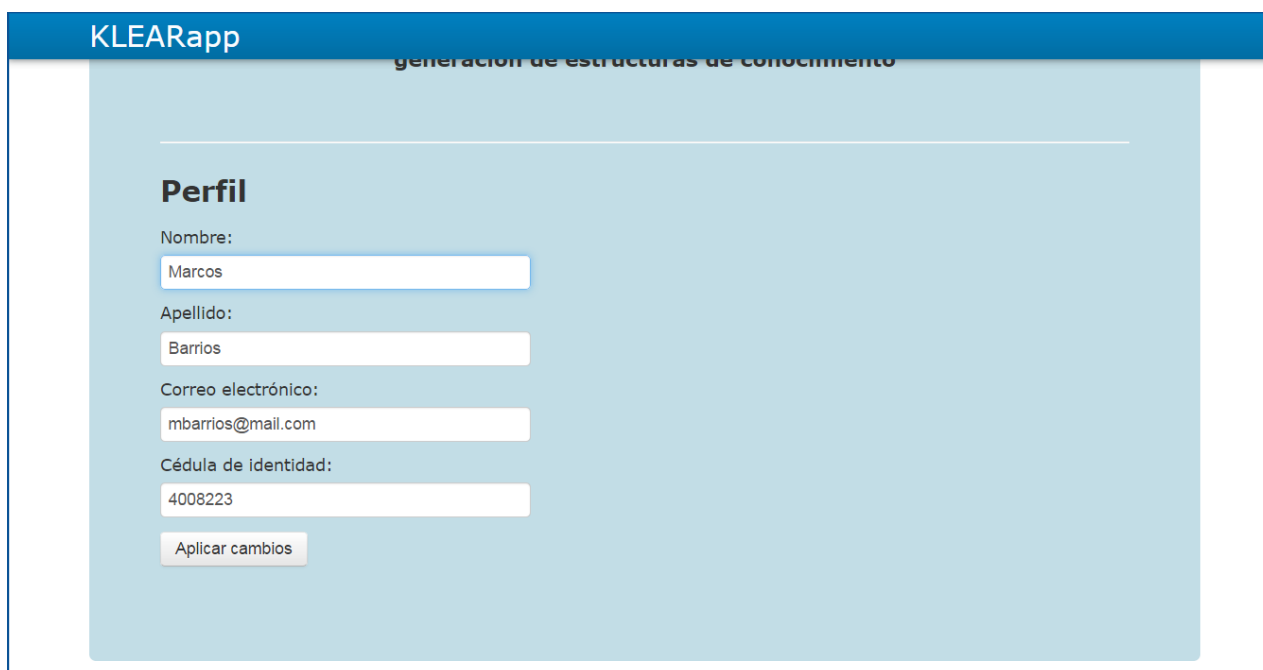
Desde la barra de navegación, al posicionarse sobre el nombre del usuario se despliega el menú con las funcionalidades relacionadas con el manejo de usuario.

Para cambiar el perfil:

1. Hacer clic en la opción **“Perfil”**.



2. El alumno puede sobrescribir aquellos campos que desea modificar.



KLEARapp generación de estructuras de conocimiento

Perfil

Nombre:

Apellido:

Correo electrónico:

Cédula de identidad:

3. Una vez hecho los cambios, hacer clic en **“Aplicar cambios”**.

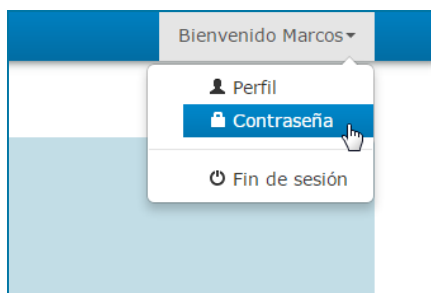
KLEARapp

Barra de navegación (continuación)

Cambiar contraseña

Para cambiar la contraseña:

1. Hacer clic en la opción “**Contraseña**”.



Complete el formulario “Cambie la contraseña” siguiendo los pasos:

1. En el campo “**Contraseña actual**”, ingresar la contraseña con que se inició sesión.
2. En “**Nueva contraseña**” ingresar la contraseña por la que se desea cambiar.
3. En donde dice “**Verifique contraseña**”, ingresar nuevamente la nueva contraseña.
4. Hacer clic en el botón “**Cambiar contraseña**”

KLEARapp

Bienvenidos a KLEARapp!

KLEARapp es una aplicación web para soporte de un sistema de escritura y generación de estructuras de conocimiento

Cambie la contraseña

Contraseña actual:

Nueva contraseña:

Verificar Contraseña:

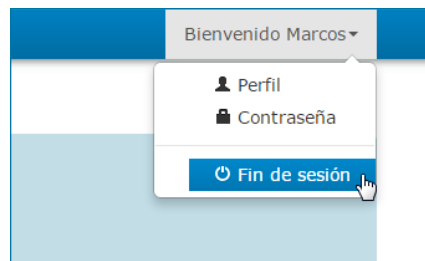
KLEARapp

Barra de navegación (continuación)

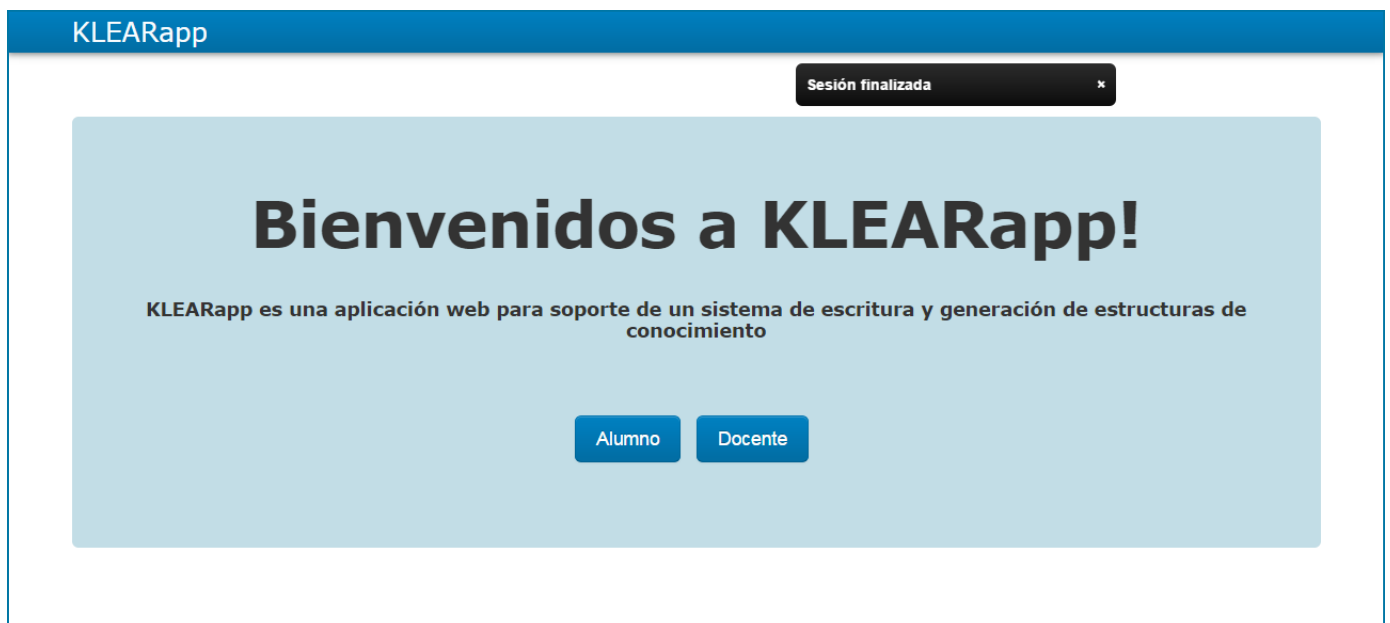
Cerrar sesión

Para cerrar sesión:

1. Hacer clic en la opción “**Fin de sesión**”.



Se redirige a la pantalla principal y se despliega en fondo negro el siguiente mensaje:



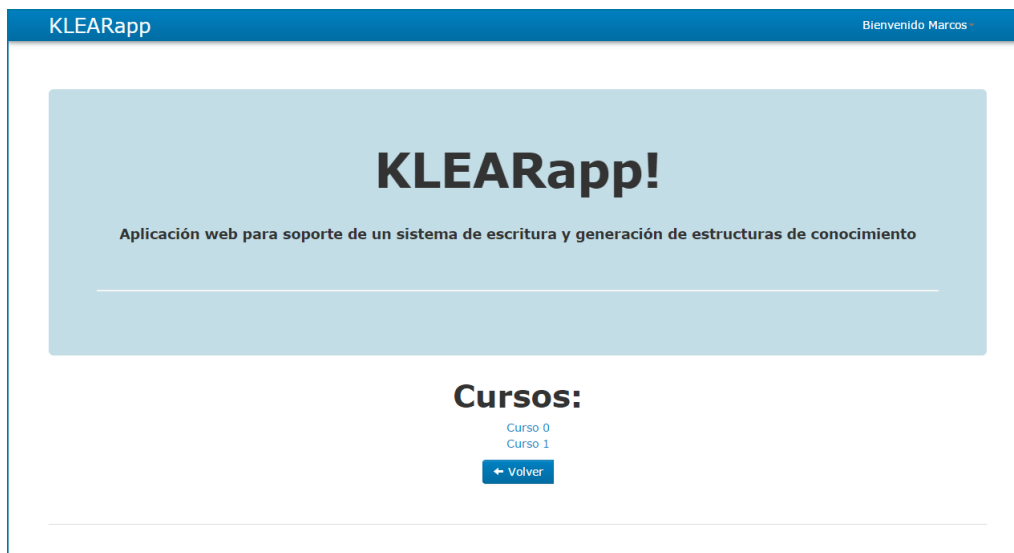
Otra forma de siempre cerrar sesión es hacer clic en el logo “**KLEARapp**”.

KLEARapp

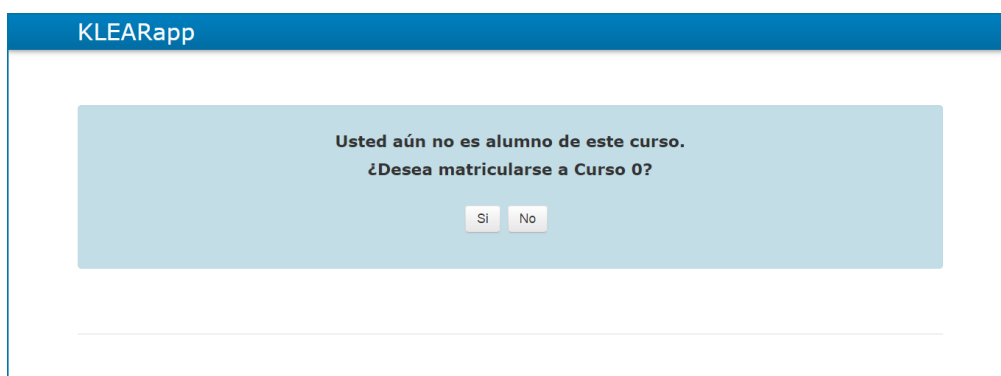
Matricularse en un curso

Para matricularse a un curso:

1. Una vez que el alumno inicia sesión, la aplicación lo dirige a la pantalla donde se listan todos los cursos. Hacer clic sobre el curso al que desea



2. La aplicación le pregunta al alumno si desea matricularse en el curso. Para confirmar dicha acción, hacer clic en el botón "Si"



En caso de seleccionar "No" la aplicación se dirige nuevamente a la pantalla donde se encuentran todos los cursos.

KLEARapp

Matricularse en un curso (continuación)

En qué estado puede encontrarse la solicitud de matriculación.

A. Matriculación pendiente

Estado “Pendiente” : Es cuando un alumno ha solicitado permiso para matricularse a un curso y está a la espera de ser aceptado o no.

Si la solicitud se encuentra aún pendiente, al intentar ingresar al curso se despliega el siguiente mensaje:



La solicitud de matriculación ya fue realizada. x

En este caso se debe esperar a que el docente acepte o rechace dicha solicitud.

B. Matriculación aceptada

Una vez que el docente ha aceptado la solicitud de la matriculación en el curso, el alumno encuentra en estado “activo” en este.

El alumno, al hacer clic en el curso, se dirige a la página principal del curso.

Estado “Activo” : Es cuando un alumno tiene permiso para acceder y administrar un curso.

KLEARapp

Matricularse en un curso (continuación)

En qué estado puede encontrarse la solicitud de matriculación (continuación).

C. Matriculación borrada

Estado "Borrado" : Es cuando se decide que el alumno seleccionado no puede tener acceso al curso.

En el caso de que el docente decida no permitir la matriculación por parte del alumno al curso, entonces una vez que intente ingresar al curso se desplegará el siguiente mensaje:

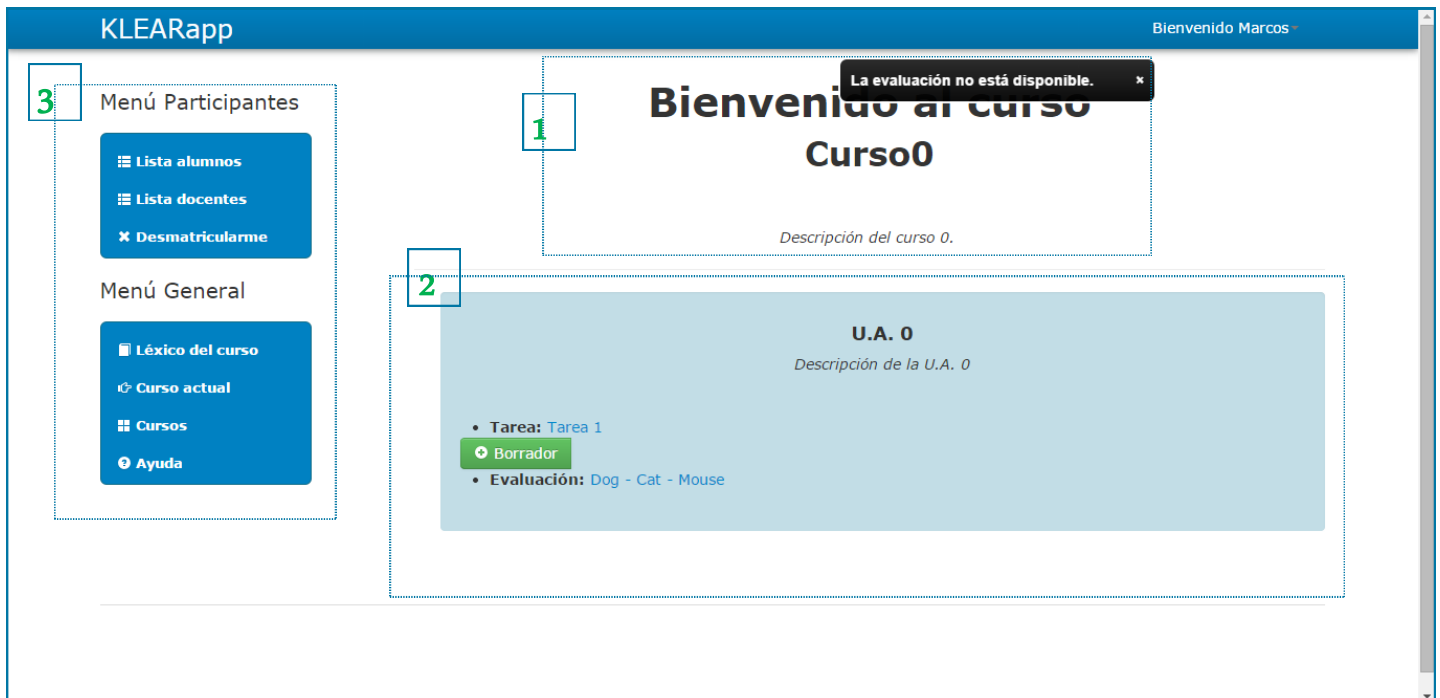


Usted está desmatriculado del curso. ✕

KLEARapp

Pantalla principal

Se muestra pantalla principal del alumno.



El alumno una vez que inicia sesión, ingresa a la pantalla principal y se encuentra con las siguientes secciones:

1. Datos generales del curso.
2. Menú Lateral:

- Menú Participantes
- Menú General

El Menú Lateral se encuentra en el lado izquierdo de la pantalla y se divide en dos secciones:

- Menú Participantes
- Menú General

KLEARapp

Menú Lateral

Menú Participantes

Lista alumnos



En esta sección el alumno puede ver la lista sus compañeros del curso.

Lista docentes

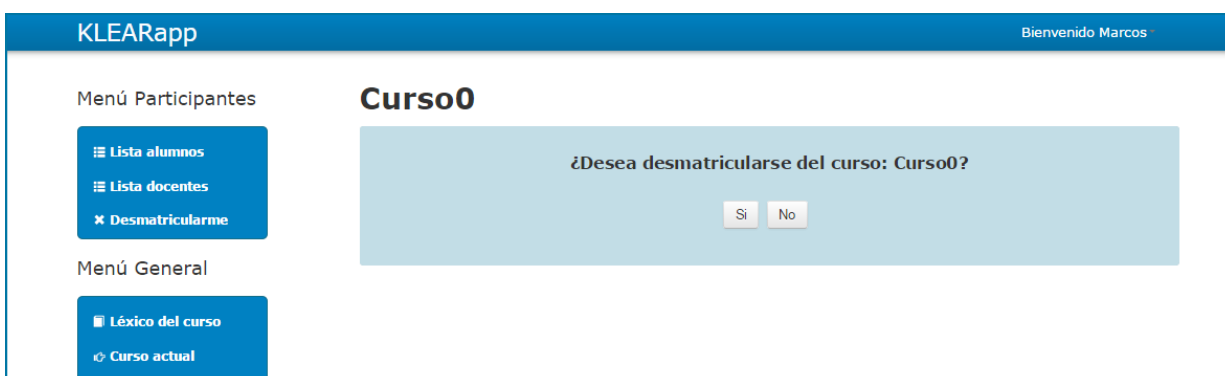


En esta sección el alumno puede ver la lista docentes del curso.

Desmatricularme



En caso que el alumno desee no participar más del curso, puede desvincularse a través de esta opción. Para eso debe elegir la opción "Sí" , en el mensaje que se despliega.

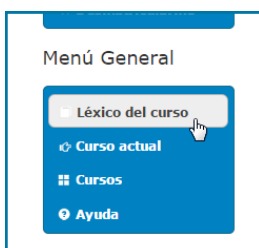


KLEARapp

Menú Lateral

Menú General

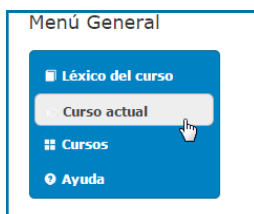
Léxico del curso



Esta sección permite agregar y modificar las categorías léxicas de las palabras que se deseen utilizar.

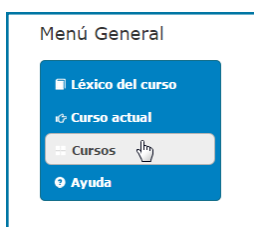
Esto fue previamente explicado en la sección [“Tarea, Corrección de léxico y reglas”](#).

Curso actual



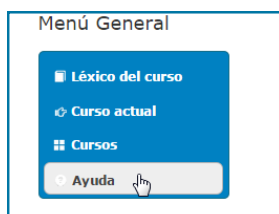
A través de este link el alumno puede dirigirse en cualquier momento al curso en el que está trabajando.

Cursos



A través de este link el alumno puede dirigirse a la pantalla con la lista de todos los cursos.

Ayuda



El alumno a través de esta sección puede descargar la ayuda.

KLEARapp

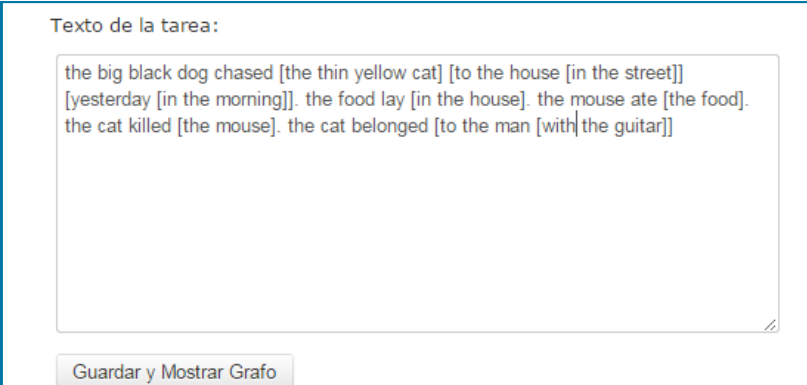
Tareas

Ingresar texto de referencia (continuación).

El **texto de referencia** es un texto que se construye en forma colaborativa por alumnos y docentes. La consigna de este texto es indicada por parte del docente en la descripción de la Tarea, que se incluye en la unidad de aprendizaje asociada.

Para ingresar el texto:

1. Seleccionar la Tarea en la que se desea ingresar el texto.
2. Ingresar el texto en el área editable bajo el título “**Texto de la tarea**”.



Texto de la tarea:

the big black dog chased [the thin yellow cat] [to the house [in the street]]
[yesterday [in the morning]]. the food lay [in the house]. the mouse ate [the food].
the cat killed [the mouse]. the cat belonged [to the man [with] the guitar]]

Guardar y Mostrar Grafo

3. Hacer clic en el botón “**Guardar y Mostrar Grafo**”.

ADVERTENCIA: Si el alumno abandona la pantalla sin seleccionar el botón “Guardar y Mostrar Grafo” entonces perderá toda la información ingresada.

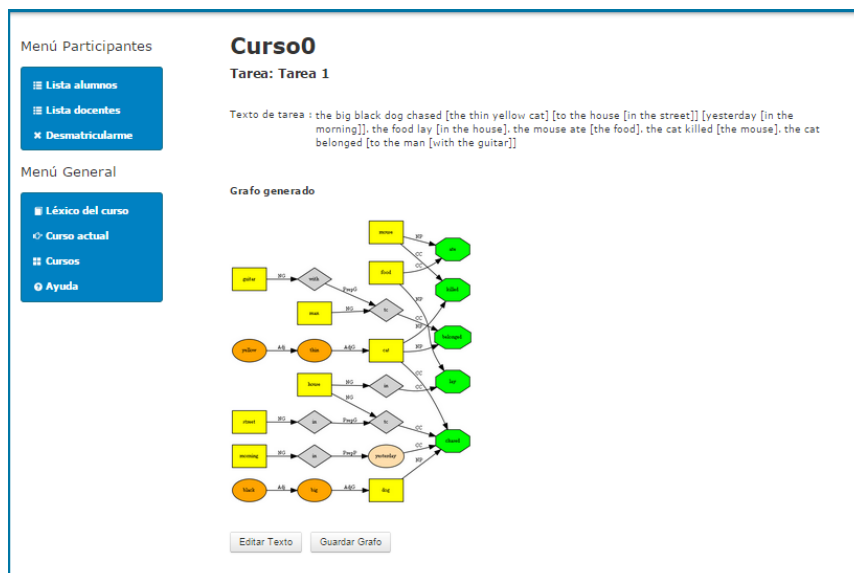
KLEARapp

Tareas

Ingresar texto de referencia (continuación).

ADVERTENCIA: Cuando una tarea ha sido “Congelada” por el Docente, la misma presenta el texto de referencia sin posibilidad de edición sino como una imagen. En este caso, el alumno puede acceder a la Tarea para estudiar el texto construido pero no podrá

- La aplicación muestra el grafo generado a partir del texto.



- Para volver a la pantalla principal hacer clic en el menú lateral, en la sección “Curso actual”.

KLEARapp

Tareas

Abrir grafo

Una vez que el alumno ha guardado su grafo puede recuperarlo. Para eso:

1. Ingresar a la Tarea en donde se guardó el grafo.
2. Hacer clic en el botón **“Grafos guardados”**.

Curso0

Tarea: Tarea 1

Nombre: Tarea 1
Descripción: Descripción de la Tarea 1

Texto de la tarea:

the big black dog chased [the thin yellow cat] [to the house [in the street]]
[yesterday [in the morning]]. the food lay [in the house]. the mouse ate [the food].
the cat killed [the mouse]. the cat belonged [to the man [with the guitar]]

✍

Se muestra en pantalla la lista con los grafos que el alumno ha guardado.

Curso0

Grafos guardados de la U.A. U.A. 0

- Grafo tarea UA 0 Curso 0 alumno
- Grafo tarea UA 0 Curso 0 Alumno
- Grafo tarea UA 0 Curso 0 Alumnoo
- Grafo 1
- Grafo 2

Para abrir el grafo simplemente se debe hacer clic sobre el Título.

KLEARapp

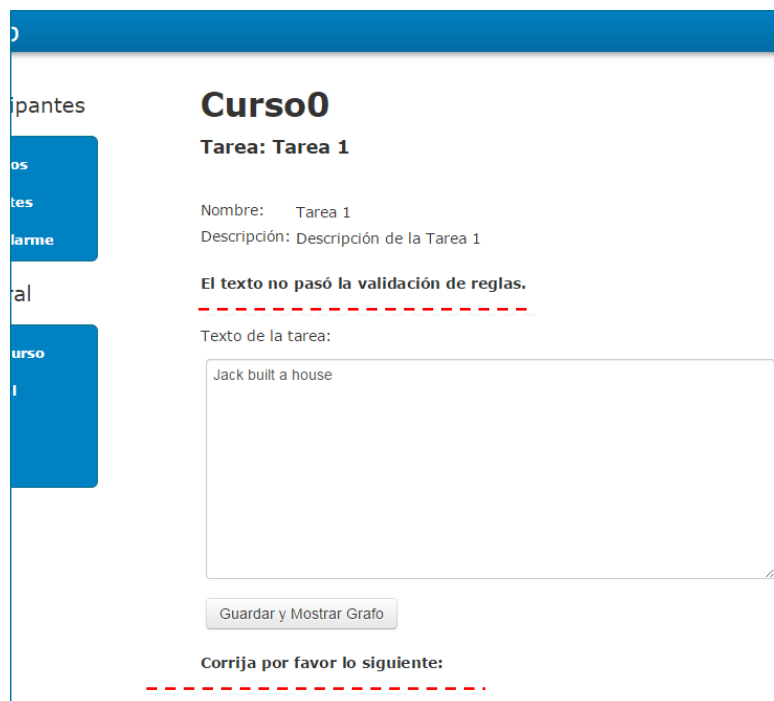
Tareas

Corrección de léxico y reglas

Puede pasar que cuando se ingrese un texto, el mismo no cumpla con las reglas del sublenguaje utilizado o que alguna palabra no pertenezca al léxico.

A. Corrección de reglas

Si al ingresar el texto, el mismo no cumple con las reglas manejadas por la aplicación, se muestra el siguiente mensaje:



The screenshot shows a web application interface for a course titled "Curso0". The main heading is "Curso0" and the task is "Tarea: Tarea 1". The task details include "Nombre: Tarea 1" and "Descripción: Descripción de la Tarea 1". A red dashed line indicates a validation error: "El texto no pasó la validación de reglas." Below this, the task text input field contains "Jack built a house". A button labeled "Guardar y Mostrar Grafo" is visible. At the bottom, another red dashed line is followed by the text "Corrija por favor lo siguiente:".

KLEARapp

Tareas

Corrección de léxico y reglas (continuación)

B. Agregar léxico

Si al ingresar el texto, el sistema no tiene algún léxico

1. Complete los datos solicitados.

Menú Participantes

- Lista alumnos
- Lista docentes
- Desmatriculame

Menú General

- Léxico del curso
- Curso actual
- Cursos
- Ayuda

Curso0

Ingrese par headword:PoS
El texto ingresado no pasó la validación de léxico.

Texto ingresado:
Mauro built [a house]

Palabra fuera del léxico
La siguiente palabra aún no se encuentra en el léxico. Asigne un par headword:PoS.

- Mauro
 - Enunciado:
 - Mauro built [a house]

Ingrese par headword:PoS para la palabra Mauro: Seleccione Po ▼

Selecione el Pos que corresponda.
(ej. En este caso "Np")

Ingresar la palabra que no se encuentra en la lista de léxico.
(ej. En este caso: "Mauro")

2. Hacer clic en el botón de más abajo, **"Continuar"**.

En caso de tener éxito la validación, se muestra el grafo correspondiente.

KLEARapp

Tareas

Corrección de léxico y reglas (continuación)

C. Modificar categoría léxica.

Si se desea modificar el léxico de una palabra ya ingresada en el sistema:

1. Ir al menú lateral, en la sección “Menú General” en la parte que dice “Léxico del curso”.



2. Ingresar en el área del buscador la palabra cuya categoría léxica se desea modificar.



KLEARapp

Tareas

Corrección de léxico y reglas (continuación)

C. Modificar categoría léxica (continuación)

Para eliminar la categoría de léxico de una palabra ya ingresada:

1. Ingresar la palabra en la sección “**Modificar léxico**” y seleccionar la categoría con la que fue ingresada.
2. Hacer clic en el botón “**Eliminar**”.

Luego se vuelve a la pantalla de “Lexicos del curso” y se despliega el siguiente mensaje a la derecha superior de la pantalla:

El par Mauro:Nc se eliminó del léxico para la palabra Mauro. x

Para agregar nueva categoría de léxico de una palabra ya ingresada:

1. Ingresar la palabra en el buscador nuevamente.

Busque una palabra en el léxico.

(Sensible a mayúsculas)

En la sección modificar léxico:

1. Ingresar nuevamente la palabra y seleccionar la nueva categoría.
2. Hacer clic en el botón “**Agregar**”.

Luego se vuelve a la pantalla de “Lexicos del curso” y se despliega el siguiente mensaje a la derecha superior de la pantalla:

El par Mauro:Np se agregó al léxico para la palabra Mauro. x

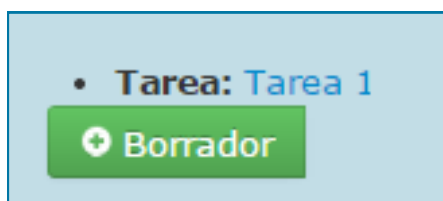
KLEARapp

Borradores

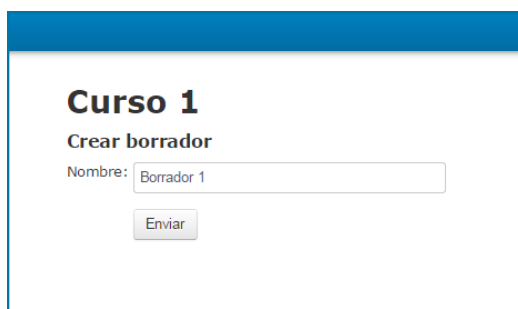
Los alumnos pueden crear **borradores** en cada unidad de aprendizaje. Los borradores son instancias donde el alumno puede ingresar texto y generar sus grafos. Son de personales por lo que sólo el mismo puede ver su información

Crear borrador.

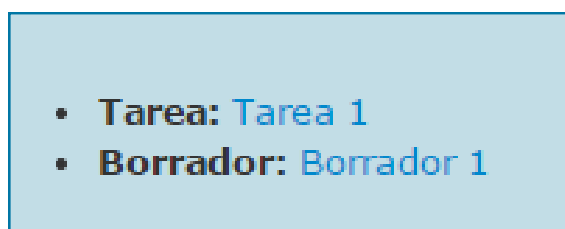
1. Hacer clic en el botón “**Borrador**”.



2. La aplicación despliega un formulario para completar con el nombre que se quiere asignar al nuevo borrador.

A screenshot of a form titled 'Curso 1' and 'Crear borrador'. It has a text input field labeled 'Nombre:' with the value 'Borrador 1' entered. Below the input field is a button labeled 'Enviar'.

3. Hacer clic en el botón “**Enviar**”.
4. Se muestra nuevamente la pantalla principal del curso donde se visualiza el enlace al borrador creado.




KLEARapp

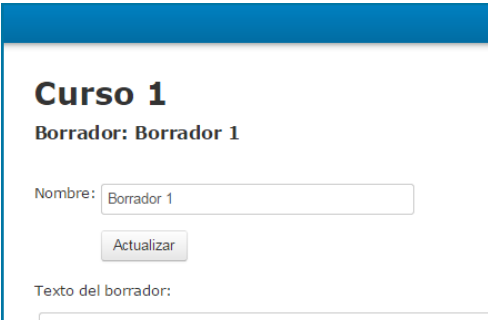
Borradores

Modificar nombre del borrador.

1. Hacer clic en el botón **“Borrador”**.

- 
- **Tarea:** Tarea 1
 - **Borrador:** Borrador 1

2. Seleccionar el campo **“Nombre”** y sobrescribir con el nuevo texto.
3. Hacer clic en el botón **“Actualizar”**.



Curso 1

Borrador: Borrador 1

Nombre:

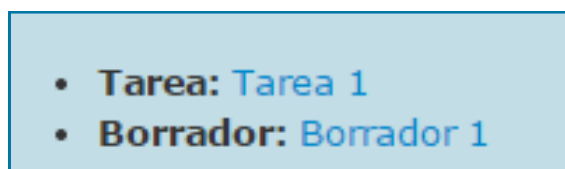
Texto del borrador:

KLEARapp

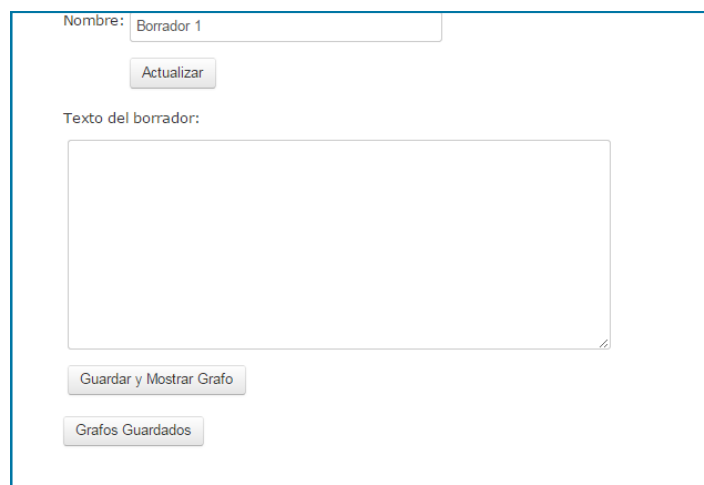
Borradores

Ingresar texto al borrador.

1. Hacer clic en el botón **“Borrador”**.



2. Ingresar el texto en el área de edición debajo del título que dice **“Texto del borrador”**.
3. Hacer clic en **“Guardar y Mostrar Grafo”**. De esta forma se guarda el texto

A screenshot of a web form for creating a draft. At the top, there is a text input field labeled "Nombre:" containing the text "Borrador 1". Below this field is a button labeled "Actualizar". Underneath is another text input field labeled "Texto del borrador:" which is currently empty. Below the text area are two buttons: "Guardar y Mostrar Grafo" and "Grafos Guardados".

ADVERTENCIA: Si el alumno abandona la pantalla sin seleccionar el botón **“Guardar y Mostrar Grafo”** entonces perderá toda la información ingresada.

Para guardar y ver los grafos guardados ver sección Tarea—guardar grafo y abrir grafo.

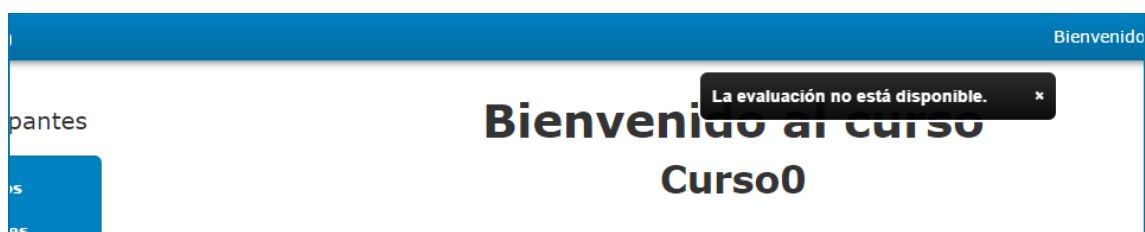
KLEARapp

Evaluación

Los docentes pueden plantear instancias de evaluación en base a los **textos de referencia** .

Acceder a la evaluación

El alumno podrá acceder a la evaluación únicamente cuando el Docente la habilite. En caso contrario, al intentar acceder aparecerá el siguiente mensaje:



En caso de estar habilitada, hacer clic en el enlace de la Evaluación.

A screenshot of the KLEARapp interface. The top bar says 'KLEARapp' and 'Bienvenido Marcos'. On the left, there are two menu sections: 'Menú Participantes' with options 'Lista alumnos', 'Lista docentes', and 'Desmatricularme'; and 'Menú General' with options 'Léxico del curso', 'Curso actual', 'Cursos', and 'Ayuda'. The main content area is titled 'Bienvenido al curso Curso0' and includes a 'Descripción del curso 0.' section. Below that, there is a 'U.A. 0' section with a 'Descripción de la U.A. 0'. A list of tasks is shown: 'Tarea: Tarea 1', 'Borrador', and 'Evaluación: Dog - Cat - Mouse'. A yellow callout box points to the evaluation task with the text: 'En este caso la evaluación se llama "Dog—Cat—Mouse"'.

KLEARapp

Evaluación

Responder evaluación

Una vez que el alumno ingresa a la evaluación, se muestran las preguntas.

Para responder:

1. Hacer clic en el enlace de la pregunta a responder.

KLEARapp

Curso0

Evaluación: Dog - Cat - Mouse

Preguntas

Pregunta 1

- Whose is the cat? (Aún sin responder)

Enviar evaluación

Se puede observar que la aplicación indica al lado derecho si la pregunta ya se encuentra respondida.

ADVERTENCIA: Una vez que se selecciona el botón “Enviar evaluación” se termina la posibilidad de editar la respuestas, dando por terminada la instancia de evaluación.

KLEARapp

Evaluación

Responder evaluación (continuación)

Al ingresar a contestar la pregunta, se muestra la siguiente pantalla:

1. Ingresar el texto que responda la pregunta.

KLEARapp

Curso0
Evaluación: Dog - Cat - Mouse

Datos de la evaluación

Whose is the cat?

Pregunta a contestar

Ingresar respuesta:

Área para ingresar respuesta

Validar

Volver a la lista de preguntas

2. En necesario verificar si el texto ingresado está bien escrito según léxico y reglas. ([Ver Sección Corrección de léxicos y reglas en Tarea](#))

KLEARapp

Evaluación

Responder evaluación (continuación)

Una vez validada la respuesta y completada,

2. Hacer clic en “Volver a la lista de preguntas”

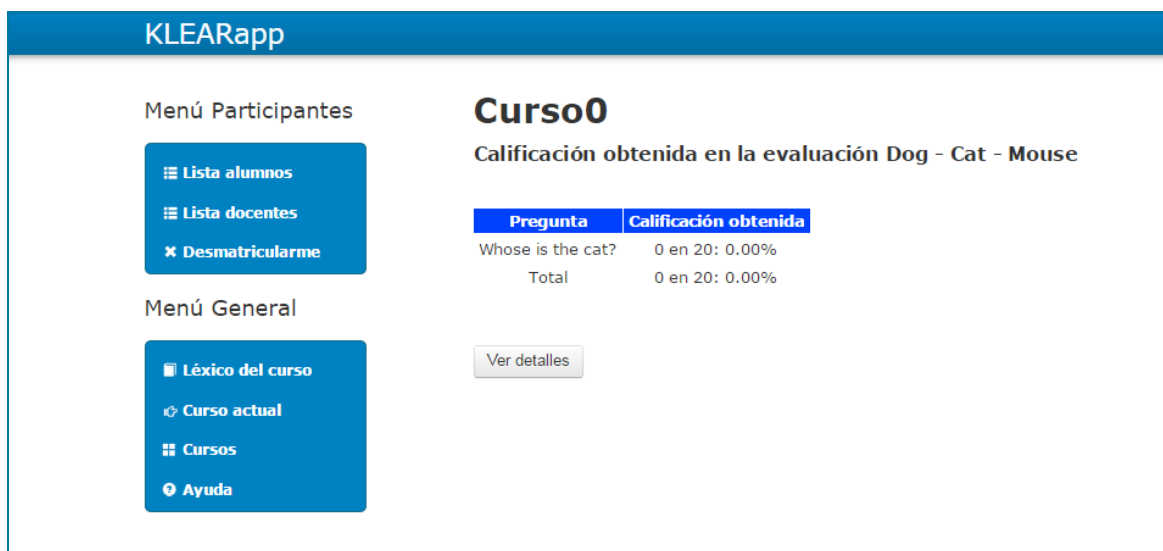
Repetir los pasos 1 y 2 con todas las preguntas.

Una vez completadas todas las preguntas,

3. Hacer clic en el botón “**Enviar**”.

Ver calificación de la evaluación

Una vez completada la evaluación, la aplicación muestra las calificaciones del alumno.



The screenshot displays the KLEARapp interface. On the left, there are two menu sections: 'Menú Participantes' with options 'Lista alumnos', 'Lista docentes', and 'Desmatricularme'; and 'Menú General' with options 'Léxico del curso', 'Curso actual', 'Cursos', and 'Ayuda'. The main content area is titled 'Curso0' and shows 'Calificación obtenida en la evaluación Dog - Cat - Mouse'. Below this is a table with two columns: 'Pregunta' and 'Calificación obtenida'. The table contains one row for the question 'Whose is the cat?' with a score of '0 en 20: 0.00%' and a 'Total' row with a score of '0 en 20: 0.00%'. A 'Ver detalles' button is located below the table.

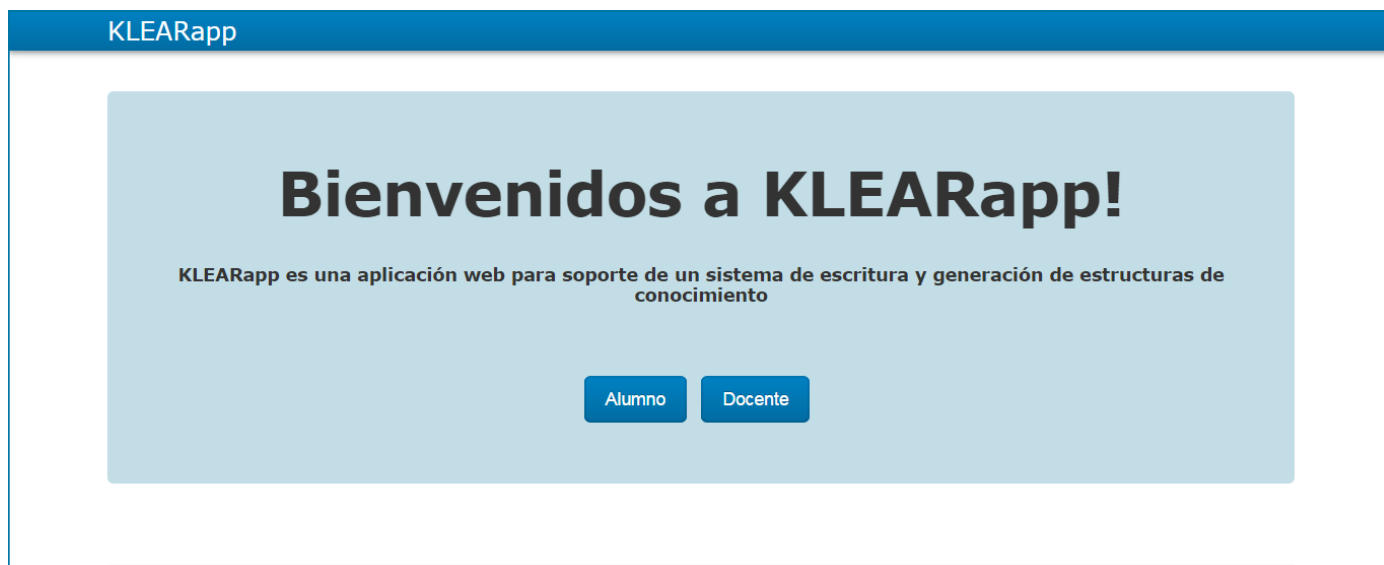
Pregunta	Calificación obtenida
Whose is the cat?	0 en 20: 0.00%
Total	0 en 20: 0.00%

Docente

KLEARapp

Registrarse como docente en la aplicación KLEARapp

1. Hacer clic en el botón “**Docente**” de la pantalla principal.



2. Hacer clic en el botón “**Registrarse**” .



KLEARapp

Registrarse como docente en la aplicación KLEARapp (continuación)

3. Completar el formulario con sus datos. Luego hacer clic en el botón “Regístrese”.

KLEARapp

Bienvenidos a KLEARapp!

KLEARapp es una aplicación web para soporte de un sistema de escritura y generación de estructuras de conocimiento

Regístrese

Nombre:

Apellido:

Correo electrónico:

Cédula de identidad:

Contraseña:

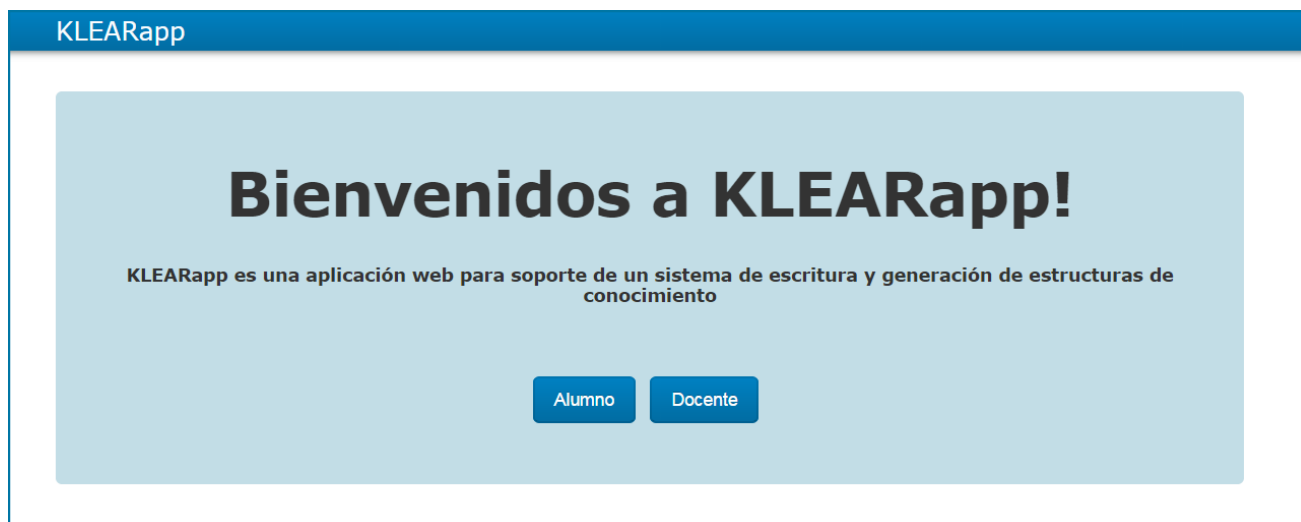
Verificar Contraseña:
 Por favor, ingrese la contraseña nuevamente

Luego se muestra la pantalla donde se listan todos los cursos que se han creado.

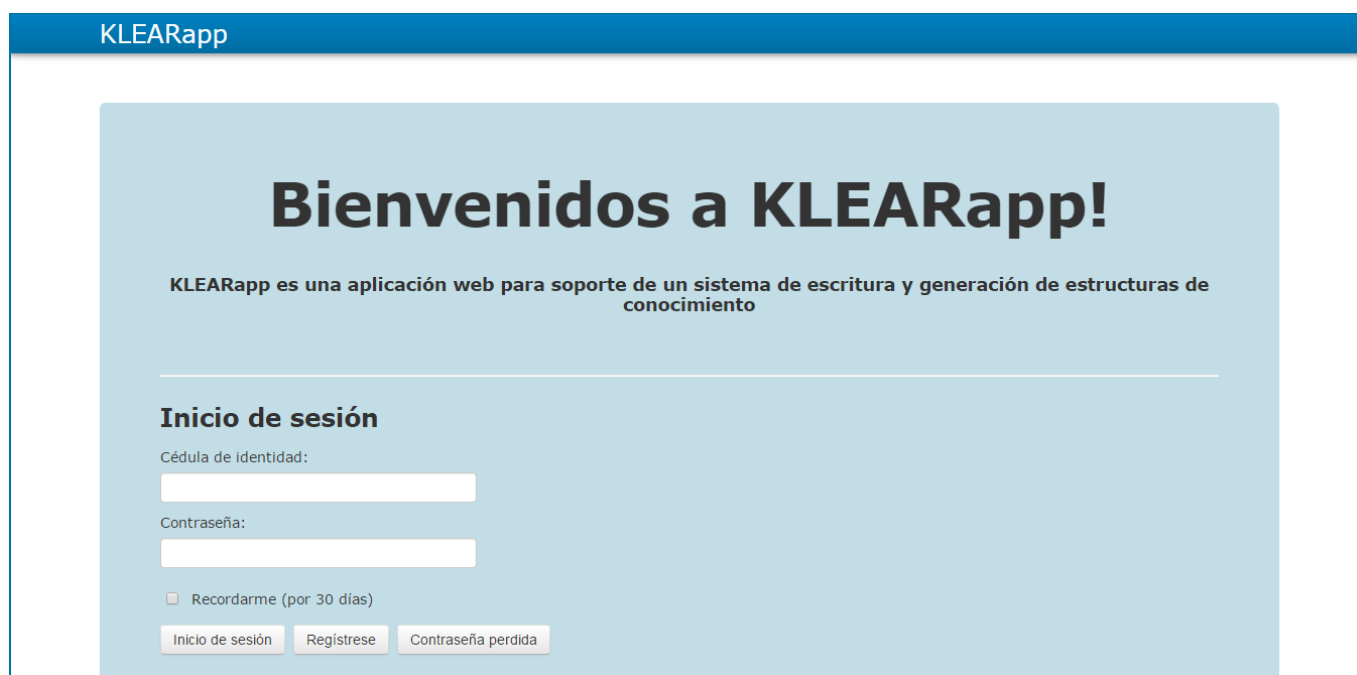
KLEARapp

Iniciar sesión

1. Hacer clic en el botón “**Docente**”.



2. Hacer clic en el botón “**Inicio de sesión**”.



KLEARapp

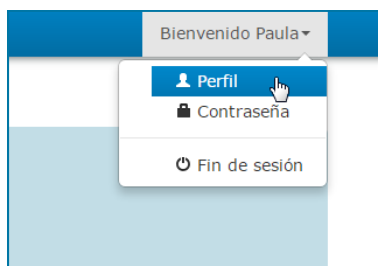
Barra de navegación

Cambiar perfil

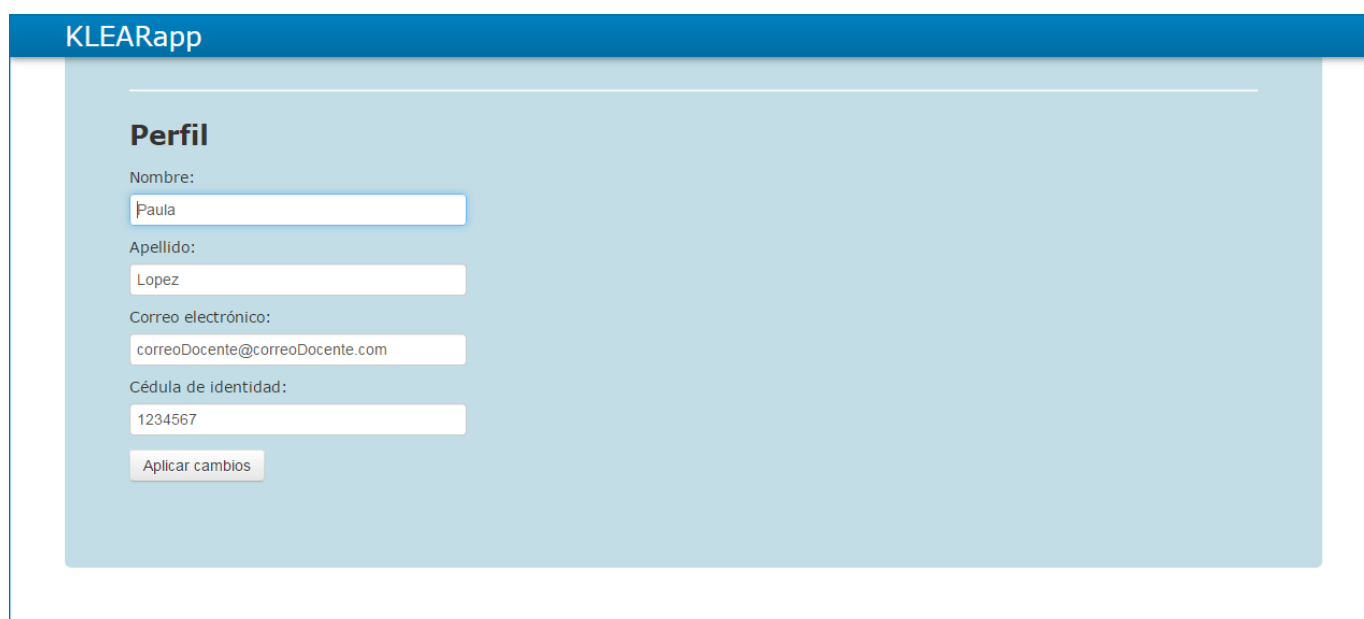
Desde la barra de navegación, al posicionarse sobre el nombre del usuario se despliega el menú con las funcionalidades relacionadas con el manejo de usuario.

Para cambiar el perfil:

1. Hacer clic en la opción **“Perfil”**.



2. El docente puede sobrescribir aquellos campos que desea modificar.



KLEARapp

Perfil

Nombre:

Apellido:

Correo electrónico:

Cédula de identidad:

3. Una vez hecho los cambios, hacer clic en **“Aplicar cambios”**.

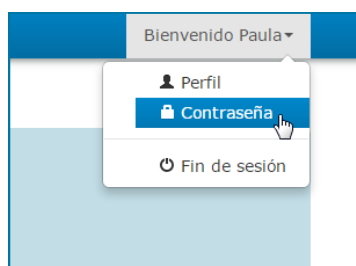
KLEARapp

Barra de navegación (continuación)

Cambiar contraseña

Para cambiar el perfil:

1. Hacer clic en la opción “**Contraseña**”.



Complete el formulario “Cambie la contraseña” siguiendo los pasos:

1. En el campo “**Contraseña actual**”, ingresar la contraseña con que se inició sesión.
2. En “**Nueva contraseña**” ingresar la contraseña por la que se desea cambiar.
3. En donde dice “**Verifique contraseña**”, ingresar nuevamente la nueva contraseña.
4. Hacer clic en el botón “**Cambiar contraseña**”

KLEARapp

Bienvenidos a KLEARapp!

KLEARapp es una aplicación web para soporte de un sistema de escritura y generación de estructuras de conocimiento

Cambie la contraseña

Contraseña actual:

Nueva contraseña:

Verificar Contraseña:

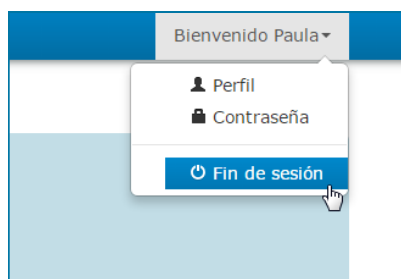
KLEARapp

Barra de navegación (continuación)

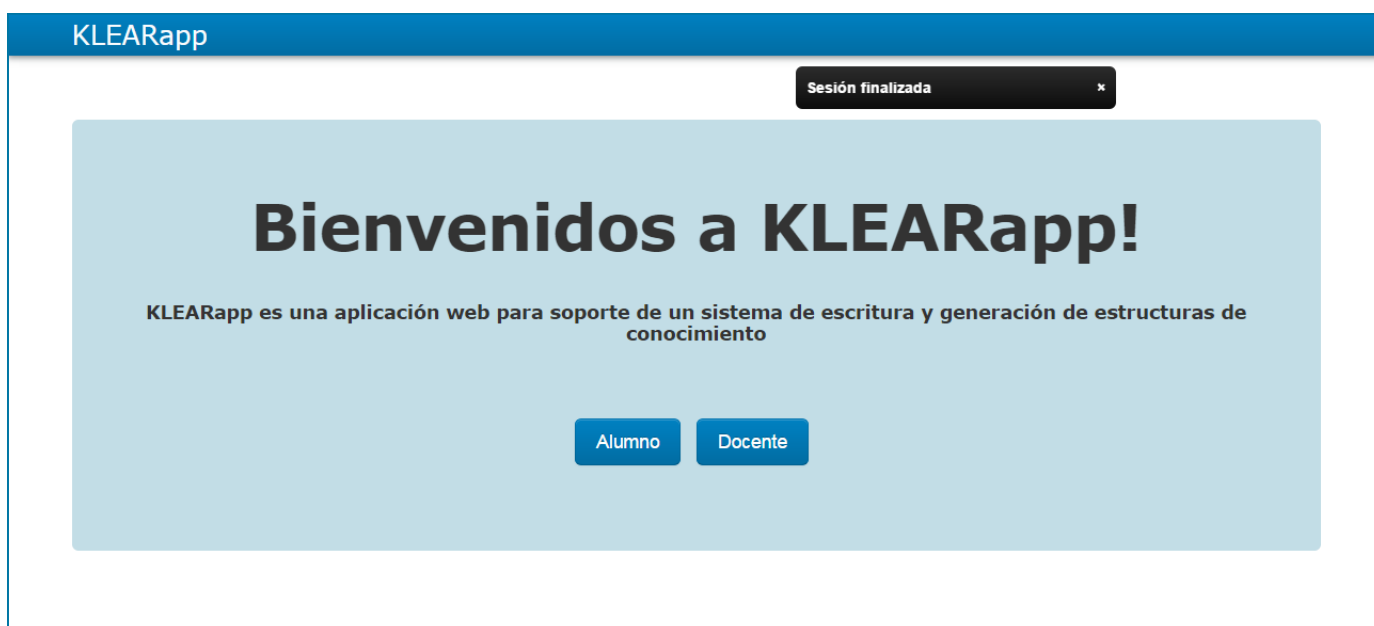
Cerrar sesión

Para cerrar sesión:

1. Hacer clic en la opción “**Fin de sesión**”.



Se redirige a la pantalla principal y se despliega en fondo negro el siguiente mensaje:

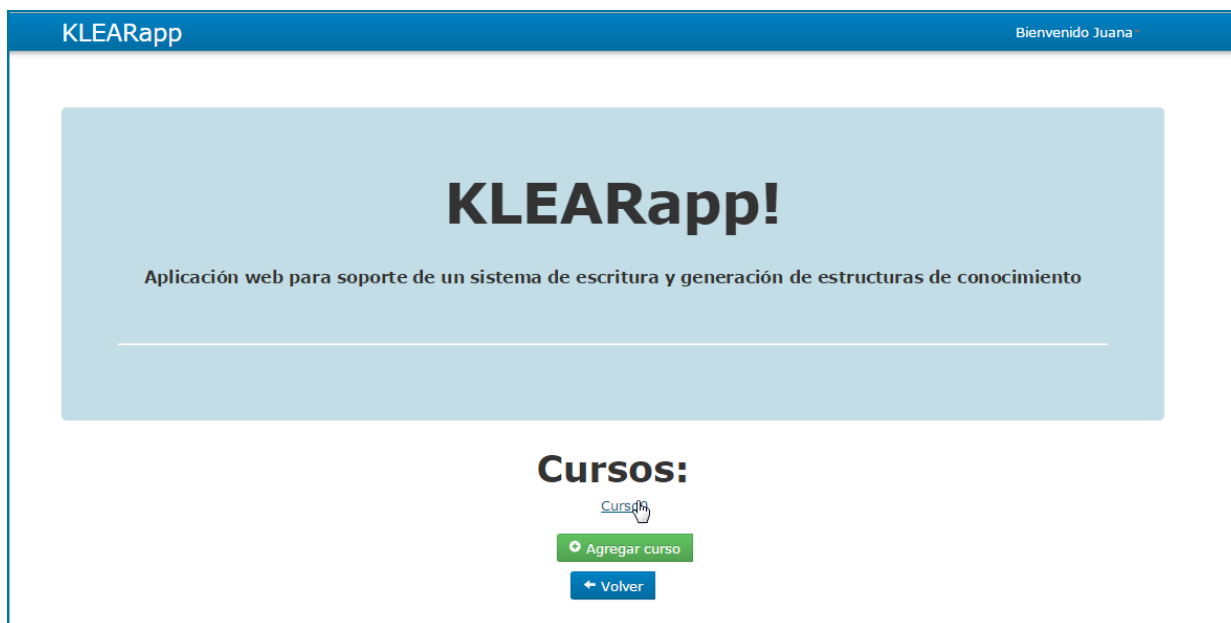


Otra forma de siempre cerrar sesión es hacer clic en el logo “**KLEARapp**”.

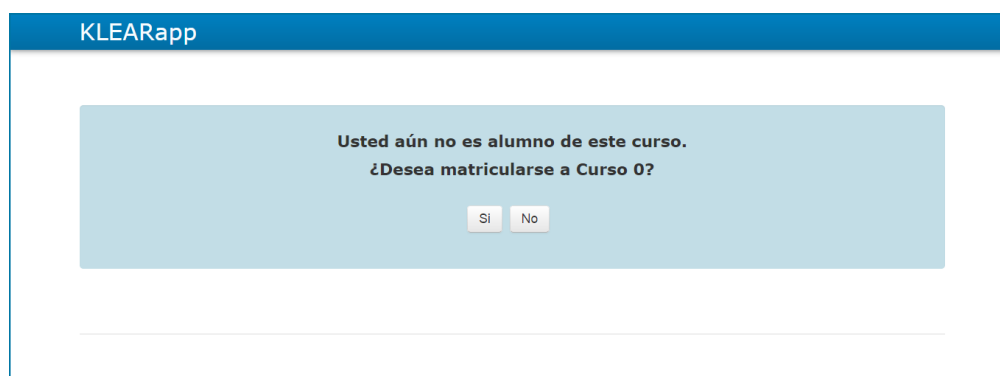
KLEARapp

Matricularse en un curso

Para matricularse a un curso:



2. La aplicación le pregunta al docente si desea matricularse en el curso. Para confirmar dicha acción, hacer clic en el botón "Si"



En caso de seleccionar "No" la aplicación se dirige nuevamente a la pantalla donde se encuentran todos los cursos.

KLEARapp

Matricularse en un curso (continuación)

En qué estado puede encontrarse la solicitud de matriculación.

A. Matriculación pendiente

Estado “Pendiente” : Es cuando un docente ha solicitado permiso para matricularse a un curso y está a la espera de ser aceptado o no.

Si la solicitud se encuentra aún pendiente, al intentar ingresar al curso se despliega el siguiente mensaje:



La solicitud de matriculación ya fue realizada. x

En este caso se debe esperar a que el administrador acepte o rechace dicha solicitud.

B. Matriculación aceptada

Una vez que el administrador ha aceptado la solicitud de la matriculación en el curso, el docente encuentra en estado “activo” en este.

El docente, al hacer clic en el curso, se dirige a la página principal del curso.

Estado “Activo” : Es cuando un docente tiene permiso para acceder y administrar un curso.

KLEARapp

Matricularse en un curso (continuación)

En qué estado puede encontrarse la solicitud de matriculación (continuación).

C. Matriculación borrada

Estado "Borrado" : Es cuando se decide que el docente seleccionado no puede tener acceso al curso.

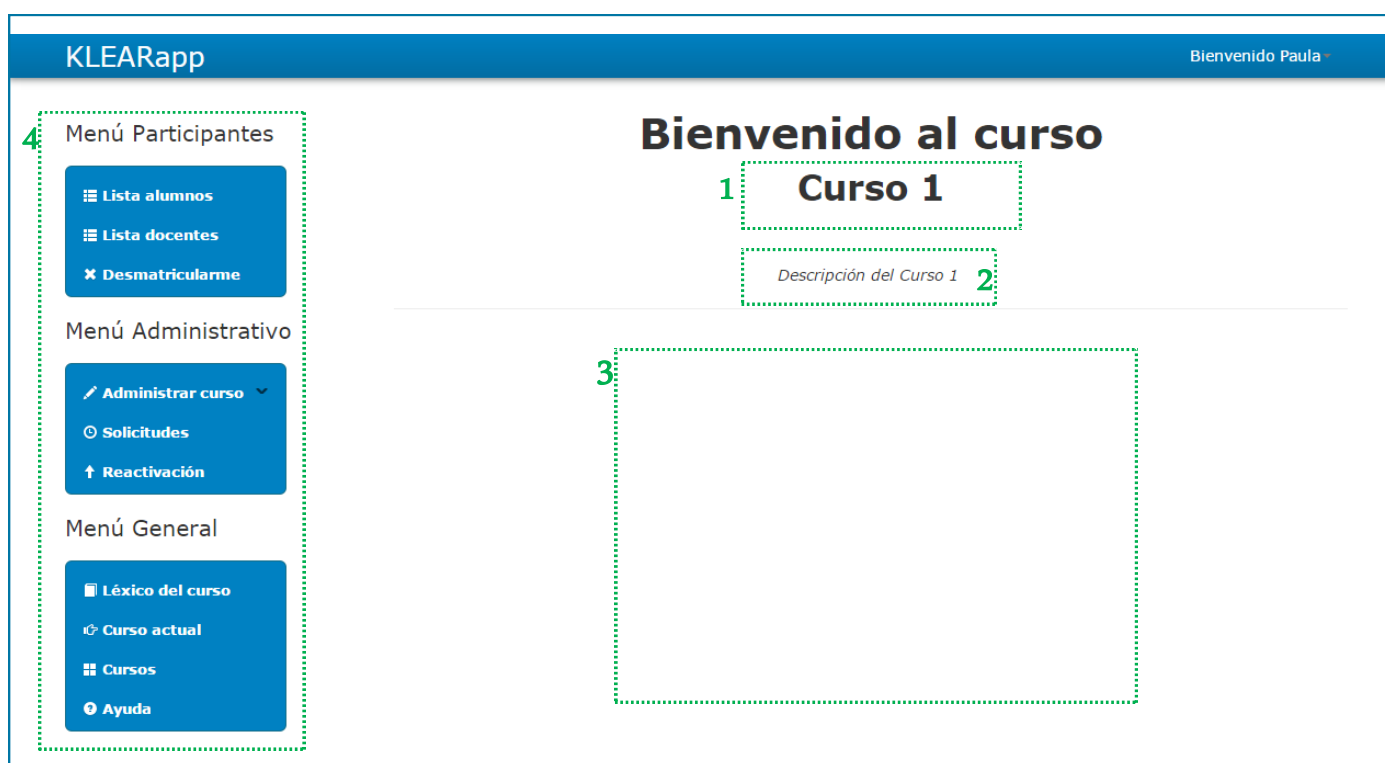
En el caso de que el administrador decida no permitir la matriculación por parte del docente al curso, entonces una vez que intente ingresar al curso se desplegará el siguiente mensaje:



Usted está desmatriculado del curso. ✕

Página principal del curso

A continuación se muestra la pantalla principal de un curso y se describen los componentes que forman al curso.



- 1- Nombre del curso.
- 2- Descripción del curso.
- 3- Área donde se encuentran todas las unidades de aprendizaje creadas.
- 4- Menú lateral donde se encuentran las herramientas del curso. Este menú a su vez se divide en las siguientes secciones:
 - [Menú Participantes](#)
 - [Menú Administrativo](#)
 - Menú General

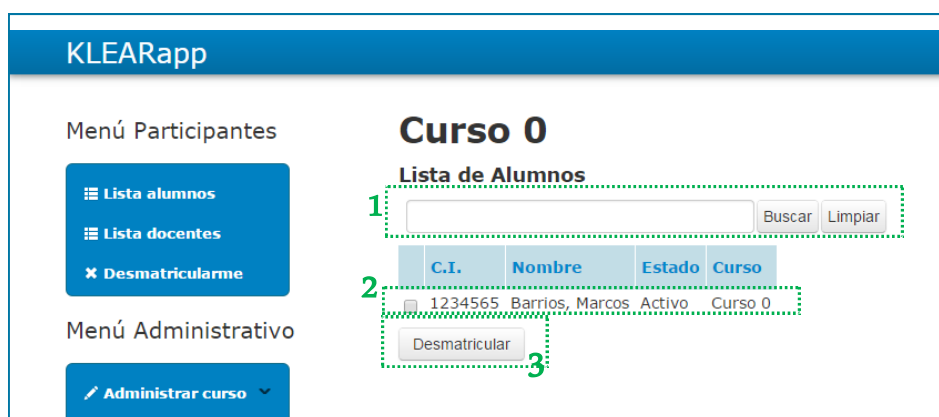
KLEARapp

Menú lateral (continuación)

En el menú lateral se puede acceder a herramientas que permiten acceder a varias funcionalidades relacionadas con el curso.

1. Menú participantes

Lista de alumnos - se listan todos los alumnos que se encuentran matriculados en el curso.



1- **Buscar un alumno.** Es un buscador que al ingresar el nombre se buscar el alumno que coincida con los datos ingresados.

2- **Datos de los alumnos.** En cada fila se describen los datos de cada alumno. Su número de cédula, su apellido y su nombre, el estado de su matriculación y a qué curso pertenece el alumno.

3- **Desmatricular a uno/varios alumnos del curso.** Se puede seleccionar varias filas de alumnos y desmatricularlos del curso haciendo clic en el botón “Desmatricular”.

KLEARapp

Menú lateral (continuación)

2. Menú Administrativo

En esta parte del menú el docente puede acceder a todas las herramientas para crear, modificar o eliminar unidades de aprendizaje, ver las solicitudes pendientes y poder reactivar alumnos previamente borrados.

Administrar curso

Modificar curso

1. Se puede modificar los datos del curso en el que se encuentra el docente. Para esto, seleccionar la opción **“Administrar curso”** del menú lateral y luego elegir la opción **“Modificar curso”**.



KLEARapp

Menú lateral (continuación)

Modificar curso (continuación)

2. Se muestra un formulario con los campos que se pueden modificar. Una vez sobrescrito los campos con los nuevos valores, hacer clic en el botón “Actualizar”. De esta forma se sobrescribe el nombre y la descripción del curso.

The screenshot displays the KLEARapp interface. At the top, the header shows 'KLEARapp' on the left and 'Bienvenido Paula' on the right. The main content area is divided into three sections: 'Menú Participantes', 'Menú Administrativo', and 'Menú General'. The 'Menú Participantes' section includes 'Lista alumnos', 'Lista docentes', and 'Desmatricularme'. The 'Menú Administrativo' section includes 'Administrar curso', 'Solicitudes', and 'Reactivación'. The 'Menú General' section includes 'Léxico del curso', 'Curso actual', 'Cursos', and 'Ayuda'. The central focus is the 'Curso 1' 'Modificar curso' form, which contains a 'Nombre' field with the value 'Curso 1' and a 'Descripción' field with the value 'Descripción del Curso 1'. An 'Actualizar' button is located below the description field.

KLEARapp

Menú lateral (continuación)

Crear U.A.

Esta opción permite al docente crear una nueva unidad de aprendizaje en el curso.

1. Hacer clic en el menú lateral en el sector “**Administración del curso**”, y seleccionar la opción “**Crear U.A.**”



2. Se despliega un formulario que permite ingresar los datos de la unidad de aprendizaje que se desea crear, hacer clic en el botón “**Enviar**”.

A screenshot of the KLEARapp interface showing the 'Curso 1' form for creating a new U.A. The interface has a blue header with 'KLEARapp' and a sidebar on the left with three menu sections: 'Menú Participantes' (with 'Lista alumnos', 'Lista docentes', and 'Desmatricularme'), 'Menú Administrativo' (with 'Administrar curso', 'Solicitudes', and 'Reactivación'), and 'Menú General' (with 'Léxico del curso'). The main content area is titled 'Curso 1' and 'Ingrese datos de la nueva U.A.'. It contains two input fields: 'Título:' and 'Descripción:'. Below the 'Descripción:' field is an 'Enviar' button.

KLEARapp

Menú lateral (continuación)

Crear U.A. (continuación)

Luego la aplicación vuelve a la pantalla principal del curso y le muestra al docente el bloque que representa a la nueva U.A.

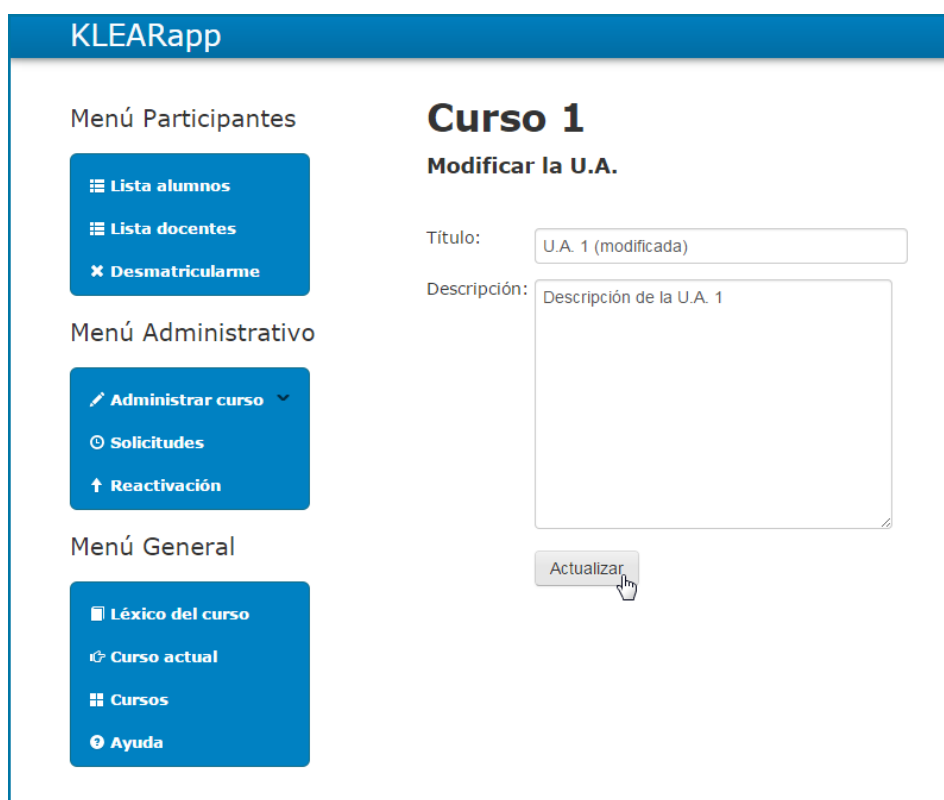
The screenshot displays the KLEARapp interface. At the top, a blue header bar contains the text 'KLEARapp' on the left and 'Bienvenido Paula' on the right. The main content area is divided into a left sidebar and a central main area. The sidebar is organized into three sections: 'Menú Participantes' with buttons for 'Lista alumnos', 'Lista docentes', and 'Desmatricularme'; 'Menú Administrativo' with buttons for 'Administrar curso', 'Solicitudes', and 'Reactivación'; and 'Menú General' with buttons for 'Léxico del curso', 'Curso actual', 'Cursos', and 'Ayuda'. The main area features a large heading 'Bienvenido al curso' followed by 'Curso 1'. Below this, there is a section titled 'Descripción del Curso 1' which contains a light blue box with the heading 'U.A. 1' and 'Descripción de la U.A. 1'. At the bottom of this box are three green buttons labeled 'Tarea', 'Borrador', and 'Evaluación'.

KLEARapp

Menú lateral (continuación)

Modificar U.A. (continuación)

3. Se muestran los datos actuales de la U.A. , para modificarlos se pueden seleccionar y sobrescribir. Una vez que el docente termina hace clic en el botón “**Actualizar**”.



The screenshot displays the KLEARapp interface. On the left, there is a sidebar menu with three sections: 'Menú Participantes' containing 'Lista alumnos', 'Lista docentes', and 'Desmatricularme'; 'Menú Administrativo' containing 'Administrar curso', 'Solicitudes', and 'Reactivación'; and 'Menú General' containing 'Léxico del curso', 'Curso actual', 'Cursos', and 'Ayuda'. The main content area is titled 'Curso 1' and 'Modificar la U.A.'. It features a 'Título:' field with the text 'U.A. 1 (modificada)' and a 'Descripción:' text area with the text 'Descripción de la U.A. 1'. Below the text area is an 'Actualizar' button with a mouse cursor pointing to it.

De esta forma se vuelve a la página principal del curso con los nuevos datos de la U.A. modificada.

KLEARapp

Menú lateral (continuación)

Eliminar U.A.

Esta opción permite al docente eliminar las unidades de aprendizaje que desee.

1. Hacer clic en el menú lateral en el sector “**Administración del curso**”, y seleccionar la opción “**Eliminar U.A.**”



2. Se debe seleccionar la U.A. que se quiere eliminar ya hacer clic en el botón “**Eliminar**”. De esta forma quedan eliminadas las U.A.’s seleccionadas.



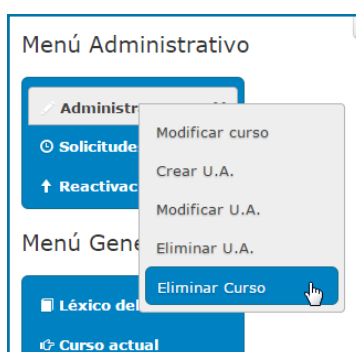
KLEARapp

Menú lateral (continuación)

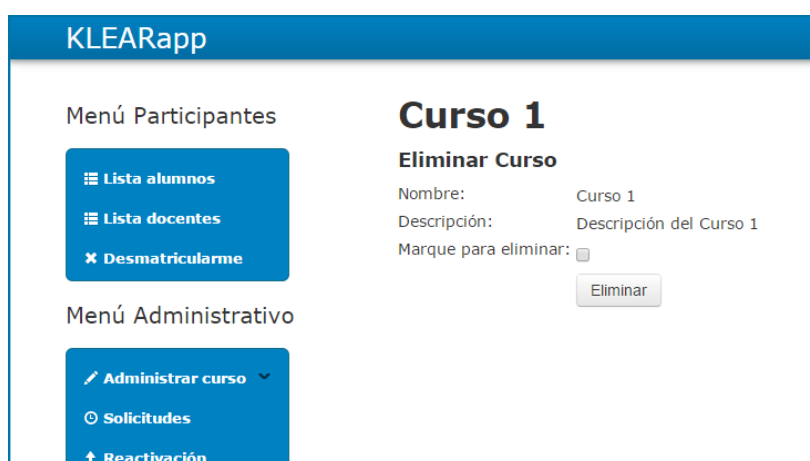
Eliminar curso

Esta opción permite al docente eliminar todos los recursos del curso creados hasta el momento.

1. Hacer clic en el menú lateral en el sector “**Administración del curso**”, y seleccionar la opción “**Eliminar curso**”



2. Hacer clic en el botón “**Eliminar**”.



KLEARapp

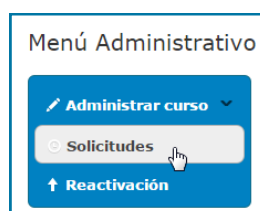
Menú lateral (continuación)

Solicitudes

Estado de las solicitudes de matriculación

En esta parte del menú el docente puede acceder a la lista de matriculaciones al curso que fueron emitidas por los alumnos. La matriculación al curso se puede aceptar o rechazar. Para aceptar una solicitud el alumno debe estar activo, de lo contrario se borra esta solicitud y de esta forma se rehúsa su matriculación.

1. Seleccionar la opción “**Solicitudes**” del menú lateral.



Se despliega una lista de las solicitudes pendiente de matriculación que se tiene en el curso.

 A screenshot of the KLEARapp interface showing the 'Curso 0' page. The page is divided into three main sections: 'Menú Participantes', 'Menú Administrativo', and 'Menú General'. The 'Menú Administrativo' section is active, showing 'Solicitudes' selected. The main content area displays 'Curso 0' and 'Solicitudes Pendientes de Alumnos'. There is a search bar with 'Buscar' and 'Limpiar' buttons. Below the search bar is a table with columns: 'C.I.', 'Nombre', 'Estado', and 'Email'. The table contains one row with the following data: '1234565', 'Barrios, Marcos', 'Pendiente', and 'None'. A dropdown menu is open over the 'Estado' column, showing options: 'Activo', 'Pendiente', and 'Borrado'. The 'Pendiente' option is currently selected. Below the table is an 'Enviar' button and an 'Exportar: CSV' link.

C.I.	Nombre	Estado	Email
1234565	Barrios, Marcos	Pendiente	None

KLEARapp

Menú lateral (continuación)

Solicitudes (continuación)

- 2a. Para que la matriculación sea activada cambiar a **“Activo”** el estado.
- 2b. Para rechazar la matriculación se selecciona **“Borrado”** en el estado. De esta forma el alumno no podrá ingresar al curso.
3. Para finalizar con el proceso, hacer clic en el botón **“Enviar”**.



The screenshot displays a web interface for 'Curso0' with the title 'Solicitudes Pendientes de Alumnos'. It features a search bar with 'Buscar' and 'Limpiar' buttons. Below is a table with columns for 'C.I.', 'Nombre', 'Estado', and 'Email'. A single row shows '1173893', 'Ramirez, Pablo', 'Pendiente' (with a dropdown arrow), and 'None'. Below the table are 'Enviar' and 'Exportar: CSV' buttons.

C.I.	Nombre	Estado	Email
1173893	Ramirez, Pablo	Pendiente	None

Exportación de datos

Al hacer clic en el enlace que dice **“Exportar: CSV”** se abre una planilla Excel con los datos de la lista de solicitudes.

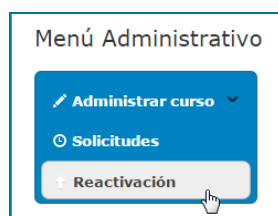
KLEARapp

Menú lateral (continuación)

Reactivación

Una vez que se borra a un alumno del curso, existe la posibilidad de incorporarlo nuevamente. Para eso se debe seguir el siguiente procedimiento:

1. Seleccionar la opción “**Reactivación**”



2. La aplicación devuelve una lista con aquellos usuarios que han sido borrados. Luego se debe seleccionar aquellos alumnos que se quiere incorporar nuevamente al curso.

A screenshot of the KLEARapp interface. The top header is blue with 'KLEARapp' on the left and 'Bienvenido Paula' on the right. The main content area is white. On the left, there are two vertical menus: 'Menú Participantes' with 'Lista alumnos', 'Lista docentes', and 'Desmatricularme'; and 'Menú Administrativo' with 'Administrar curso', 'Solicitudes', and 'Reactivación'. The 'Reactivación' option is highlighted. The main content area shows 'Curso 0' and a title 'Lista de Alumno y Docentes en estado 'Borrado' o 'Eliminado''. Below the title is a search bar with 'Buscar' and 'Limpiar' buttons. A table with columns 'C.I.', 'Nombre', 'Estado', and 'Email' is displayed. The first row shows '1425361', 'Martinez, Santiago', 'Borrado', and 'None'. There are 'Vista' and 'Editar' buttons for each row. Below the table is an 'Exportar: CSV' button.

KLEARapp

Menú lateral (continuación)

Reactivación (continuación)

3. Hacer clic en el botón “**Editar**”.
4. Luego el sistema despliega un formularios con los datos del alumno. Cambiar el estado a “**Activo**”.
5. Hacer clic en “**Enviar**”.



The screenshot displays the KLEARapp interface. On the left, there are three menu sections: 'Menú Participantes' with buttons for 'Lista alumnos', 'Lista docentes', and 'Desmatricularme'; 'Menú Administrativo' with buttons for 'Administrar curso', 'Solicitudes', and 'Reactivación'; and 'Menú General' with buttons for 'Léxico del curso' and 'Curso actual'. The main content area is titled 'Curso 0' and 'Lista de Alumno y Docentes en estado 'Borrado' o 'Eliminado''. It contains a form with the following fields: 'Identificador: 1425361', 'Email: None', 'Nombre: Martinez, Santiago', 'Curso: Curso 0', 'Rol: alumno', 'Estado: Borrado' (with a dropdown menu open showing 'Activo', 'Borrado', and 'None'), 'Inicio:', and 'Fin:'. There are '←Atrás' and 'Vista' buttons at the top, and an 'Enviar' button at the bottom.

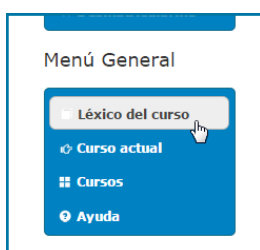
De esta forma el usuario vuelve a ser alumno del curso.

KLEARapp

Menú Lateral (continuación)

Menú General

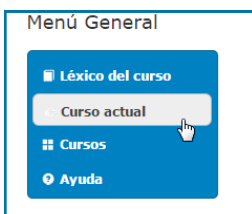
Léxico del curso



Esta sección permite agregar y modificar las categorías léxicas de las palabras que se deseen utilizar.

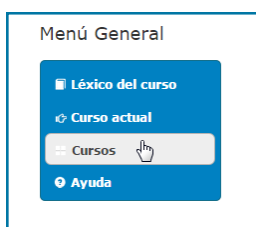
Esto fue previamente explicado en la sección [“Tarea, Corrección de léxico y reglas”](#).

Curso actual



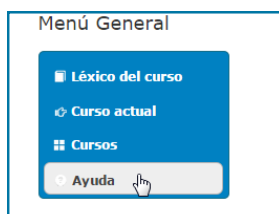
A través de este link el docente puede dirigirse en cualquier momento al curso en el que está trabajando.

Cursos



A través de este link el docente puede dirigirse a la pantalla con la lista de todos los cursos.

Ayuda



El docente a través de esta sección puede descargar la ayuda.

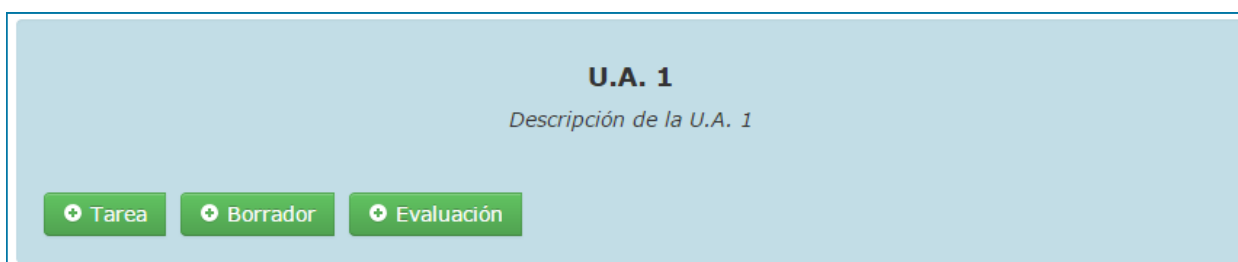
KLEARapp

Tareas

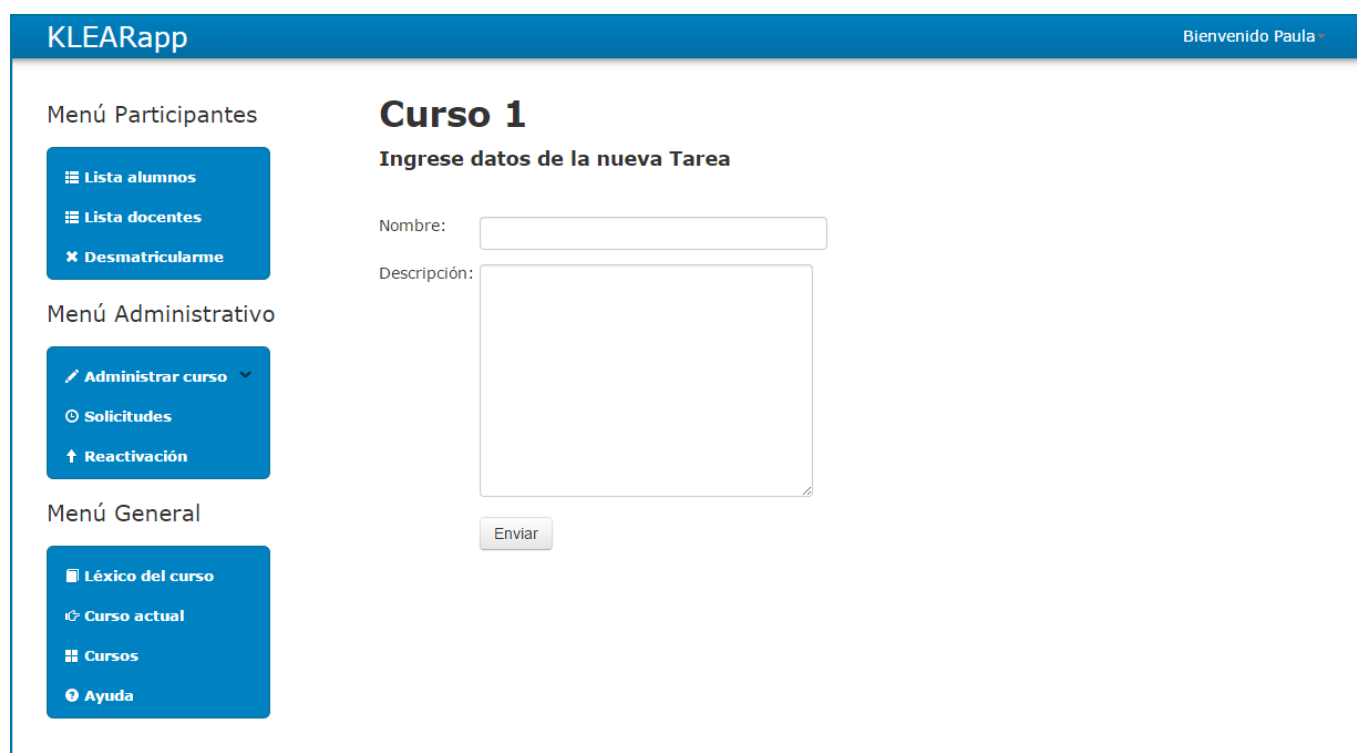
Crear Tarea

Una vez que se crea una unidad de aprendizaje, es posible crear una tarea siguiendo el siguiente procedimiento.

- 1- Hacer click en el botón “Tarea”



- 2- Se despliega un formulario para completar con los datos de la nueva tarea. Para finalizar el proceso hacer clic en “Enviar”.



KLEARapp Bienvenido Paula

Curso 1
Ingrese datos de la nueva Tarea

Nombre:

Descripción:

Menú Participantes

- Lista alumnos
- Lista docentes
- Desmatricularme

Menú Administrativo

- Administrar curso
- Solicitudes
- Reactivación

Menú General

- Léxico del curso
- Curso actual
- Cursos
- Ayuda

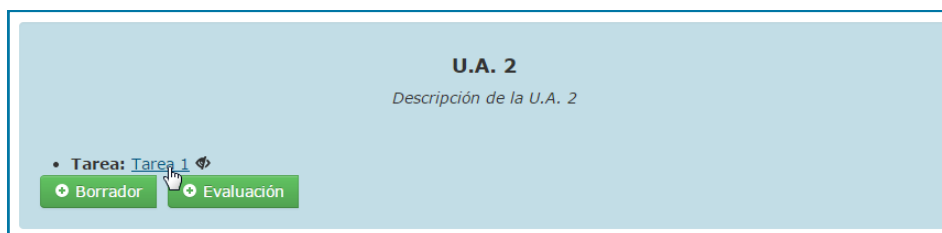
KLEARapp

Tareas

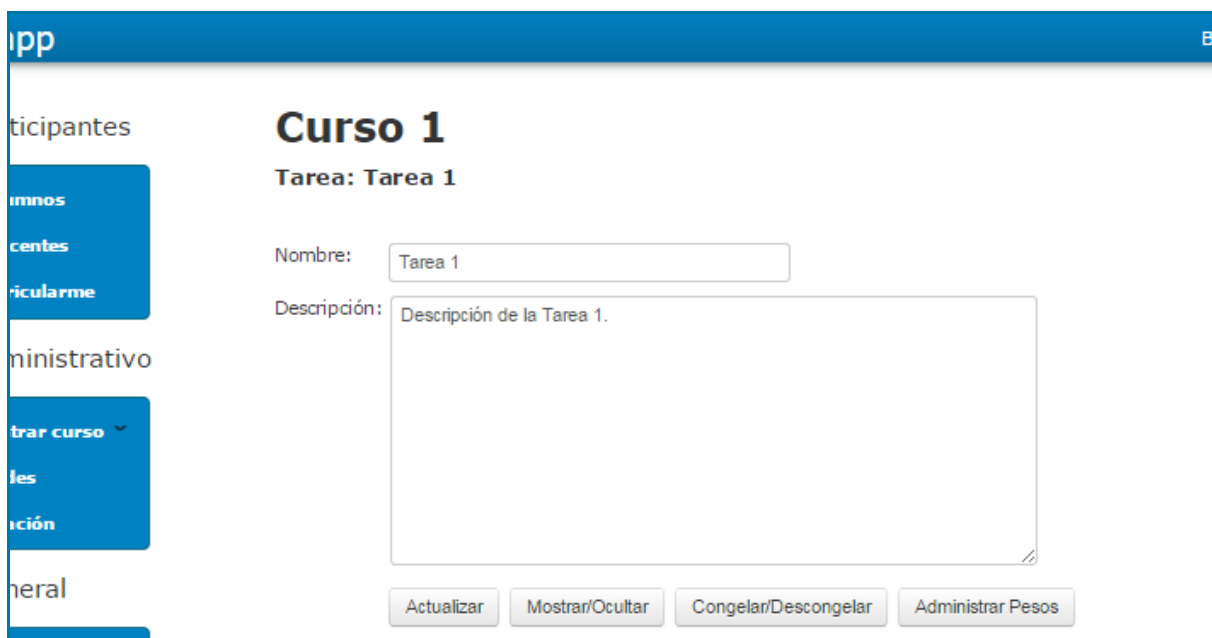
Modificar Tarea

Una vez creada la tarea se puede modificar sus datos y escribir el texto de referencia de la U.A. correspondiente.

- 1- Hacer click en el enlace de la Tarea creada.



- 2- Se despliega la pantalla de la Tarea. En la parte superior se permite modificar los datos de la tarea. Para terminar este proceso, hacer click en "Actualizar".



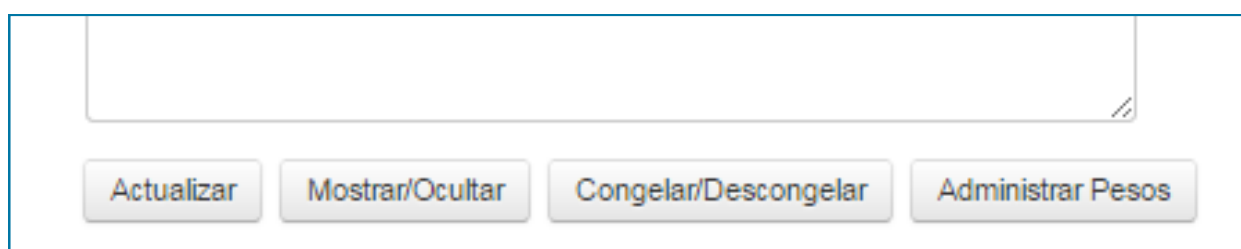
KLEARapp

Tareas

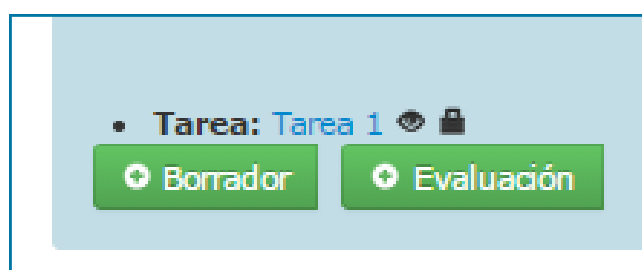
Mostrar u Ocultar Tarea

Estando ubicado en la pantalla de la Tarea el docente puede decidir ocultarla ó en caso contrario que vuelva a mostrarse. Para ambos casos:

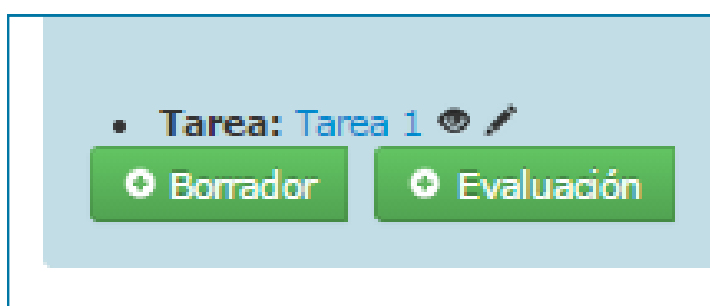
Hacer click en el botón **“Mostrar/Ocultar”**



Si la Tarea está congelada a los alumnos, se muestra con un ícono a su lado derecho con un candado. De esta forma los alumnos no podrán editar el texto de referencia.



Si la Tarea está descongelada para los alumnos, se muestra con un ícono a su lado derecho que tiene un lápiz.



KLEARapp

Tareas

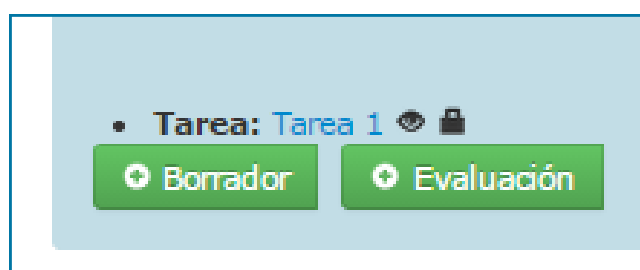
Congelar o descongelar tarea

Estando ubicado en la pantalla de la Tarea el docente puede decidir congelar ó descongelar la misma. Para ambos casos:

Hacer click en el botón **“Congelar/Descongelar”**



Si la Tarea está visible a los alumnos, se muestra con un ícono a su lado derecho que tiene la forma de un ojo abierto.



Si la Tarea está oculta a los alumnos, se muestra con un ícono a su lado derecho que tiene la forma de un ojo cerrado.

KLEARapp

Tareas

Ingresar texto de referencia.

Una vez creada a Tarea por el docente, el alumno podrá ingresar el texto de referencia.

1. Ingresar en el enlace de la Tarea a la que se desea ingresar el texto.
2. Ingresar el texto en el área editable bajo el título **“Texto de la tarea”**.

Texto de la tarea:

the big black dog chased [the thin yellow cat] [to the house [in the street]]
[yesterday [in the morning]]. the food lay [in the house]. the mouse ate [the food].
the cat killed [the mouse]. the cat belonged [to the man [with the guitar]]

Guardar y Mostrar Grafo

3. Hacer clic en el botón **“Guardar y Mostrar Grafo”**. El alumno debe asegurarse de realizar este paso para no perder la información.
4. La aplicación muestra el grafo generado a partir del texto.

KLEARapp Bienvenido Paula

Menú Participantes

- Lista alumnos
- Lista docentes
- Desmatricularme

Menú Administrativo

- Administrar curso
- Solicitudes
- Reactivación

Menú General

- Léxico del curso
- Curso actual
- Cursos
- Ayuda

curso0

Tarea: tdelau0delcurso0

Texto de tarea : the big black dog chased [the thin yellow cat] [to the house [in the street]] [yesterday [in the morning]]. the food lay [in the house], the mouse ate [the food], the cat killed [the mouse], the cat belonged [to the man [with the guitar]]

Grafo generado

Editar Texto Guardar Grafo

5. Para volver a la pantalla principal hacer clic en el menú, en la sección **“Curso actual”**.

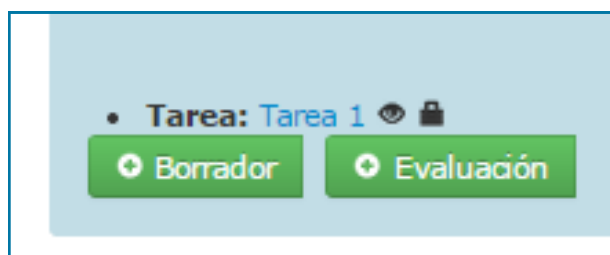
KLEARapp

Tareas

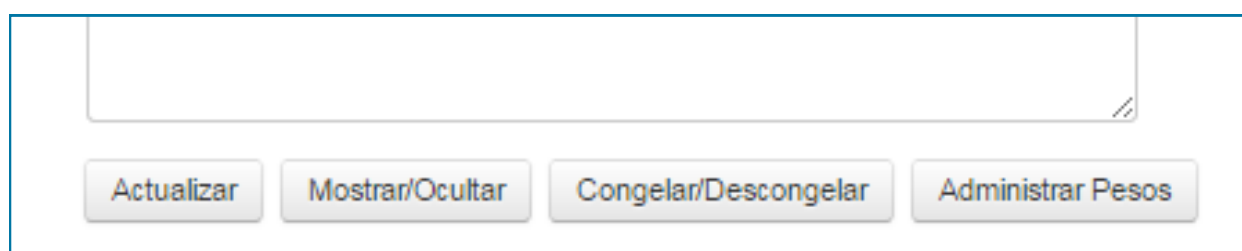
Administrar pesos

La asignación de pesos permite al docente asignarle un peso a cada nodo de acuerdo a su importancia. Para esto primero debe ingresarse el texto de referencia. Una vez que se tiene el texto se sigue el siguiente procedimiento:

1. El docente debe asegurarse que la Tarea se encuentre en estado “congelado” y visible para el alumno. Para ello verificar que se encuentre el ícono con candado al lado derecho del enlace de la Tarea, en la pantalla



2. Ingresar a la Tarea.
3. Hacer clic en el botón **“Administrar Pesos”**.

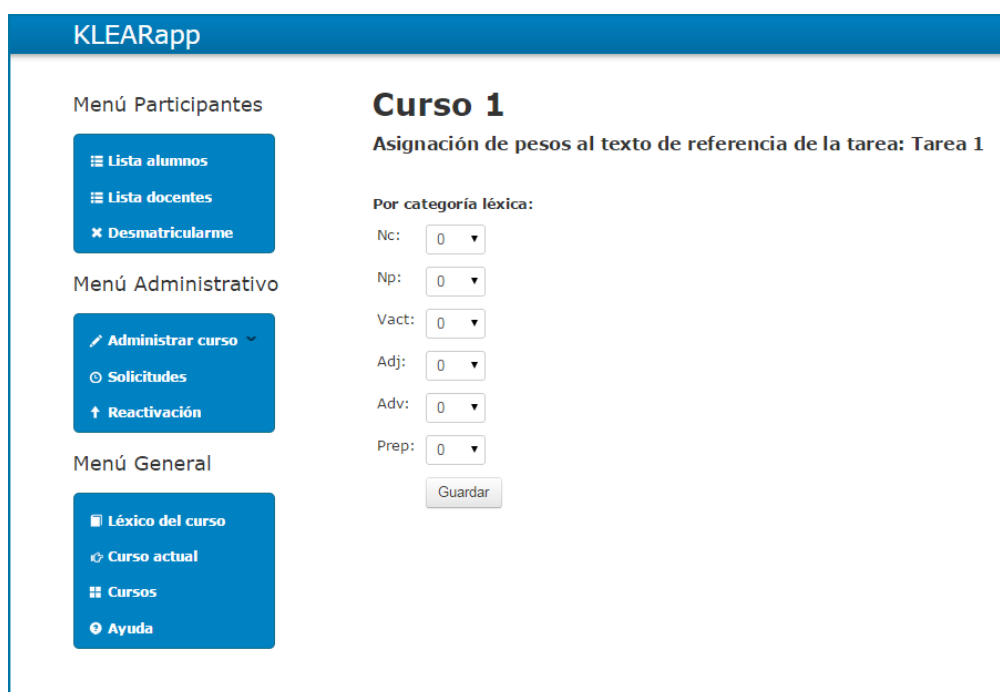


KLEARapp

Tareas

Administrar pesos (continuación)

1. Se despliega una pantalla con todas las categorías léxicas. Seleccionar un valor para cada una.



The screenshot shows the KLEARapp interface. On the left, there are three menu sections: 'Menú Participantes' with options 'Lista alumnos', 'Lista docentes', and 'Desmatricularme'; 'Menú Administrativo' with options 'Administrar curso', 'Solicitudes', and 'Reactivación'; and 'Menú General' with options 'Léxico del curso', 'Curso actual', 'Cursos', and 'Ayuda'. The main content area is titled 'Curso 1' and 'Asignación de pesos al texto de referencia de la tarea: Tarea 1'. Below this, there is a section 'Por categoría léxica:' with five dropdown menus for 'Nc:', 'Np:', 'Vact:', 'Adj:', and 'Adv:', each currently set to '0'. A 'Prep:' dropdown is also present but not fully visible. At the bottom of this section is a 'Guardar' button.

2. Una vez que el docente termina la asignación de pesos, debe hacer clic en el botón “**Guardar**”.

KLEARapp

Tareas

Administrar pesos (continuación)

3. Si se desea modificar un nodo en particular. Para eso escriba en el espacio al lado del ícono de la lupa el nodo que se desea buscar

Por categoría léxica:

Nc:

Np:

Vact:

Adj:

Adv:

Prep:

¿Desea asignar un peso especial a alguna palabra?

Busque el nodo al que quiere modificarle el peso.

(Sensible a mayúsculas)

4. Luego ingresar el nuevo valor. Y hacer clic en el botón **“Guardar peso”**.

• **big** tiene PoS: Adj y peso: . Modifique el peso si lo desea.

5. Y se despliega el siguiente mensaje.

¡Se asignó peso con éxito!
✕

KLEARapp

Tareas

Administrar pesos (continuación)

Al final de la pantalla se muestran dos botones : “**Mostrar Grafo**” y “**Mostrar Asignación**”.

• **big** tiene PoS: Adj y peso: . Modifique el peso si lo desea.

Guardar Peso

Mostrar Grafo

Mostrar Asignación

Si el docente desea ver el grafo generado a partir del texto de referencia, debe hacer clic en el botón “**Mostrar Grafo**”.

KLEARapp Bienvenido Paula

Menú Participantes

- Lista alumnos
- Lista docentes
- Desmatricularme

Menú Administrativo

- Administrar curso
- Solicitudes
- Reactivación

Menú General

- Léxico del curso
- Curso actual
- Cursos
- Ayuda

curso0

Tarea: Tarea 1

Texto de tarea : : the big black dog chased [the thin yellow cat] [to the house [in the street]] [yesterday [in the morning]]. the food lay [in the house]. the mouse ate [the food]. the cat killed [the mouse]. the cat belonged [to the man [with the guitar]]

Grafo generado

Mostrar Asignación Modificar Pesos

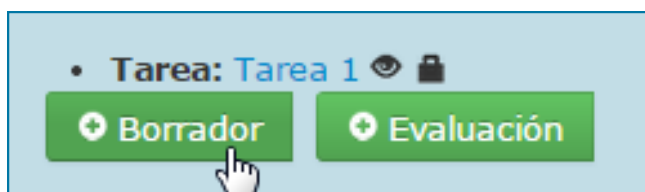
KLEARapp

Borradores

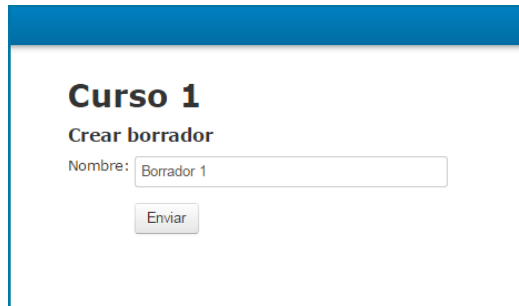
Crear borrador.

El docente tiene la posibilidad de crear un borrador en cada unidad de aprendizaje. Para esto debe seguir los siguientes pasos:

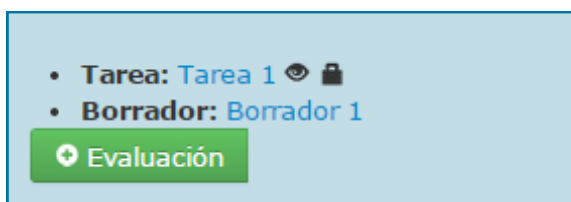
1. Hacer clic en el botón “**Borrador**”.



2. La aplicación despliega un formulario para completar con el nombre que se quiere asignar al nuevo borrador.

A screenshot of a form titled 'Curso 1' and 'Crear borrador'. It has a text input field labeled 'Nombre:' with the text 'Borrador 1' entered. Below the input field is a button labeled 'Enviar'.

3. Hacer clic en el botón “**Enviar**”.
4. Se muestra nuevamente la pantalla principal del curso donde se visualiza el enlace al borrador creado.



KLEARapp

Borradores

Modificar nombre del borrador.

1. Hacer clic en el botón **“Borrador”**.

- **Tarea:** Tarea 1
- **Borrador:** Borrador 1

2. Seleccionar el campo **“Nombre”** y sobrescribir con el nuevo texto.
3. Hacer clic en el botón **“Actualizar”**.



Curso 1
Borrador: Borrador 1

Nombre:

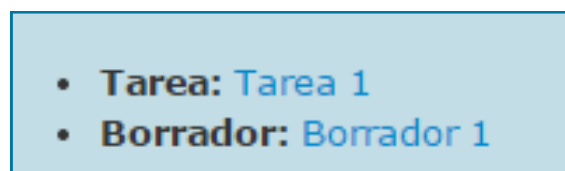
Texto del borrador:

KLEARapp

Borradores

Ingresar texto al borrador.

1. Hacer clic en el botón **“Borrador”**.



1. Ingresar el texto en el área de edición debajo del título que dice **“Texto del borrador”**.
2. Hacer clic en **“Guardar y Mostrar Grafo”**. De esta forma se guarda el texto

Nombre:

Texto del borrador:

Al hacer clic en el botón **“Grafos Guardados”** se puede acceder a todos los grafos previamente almacenados.

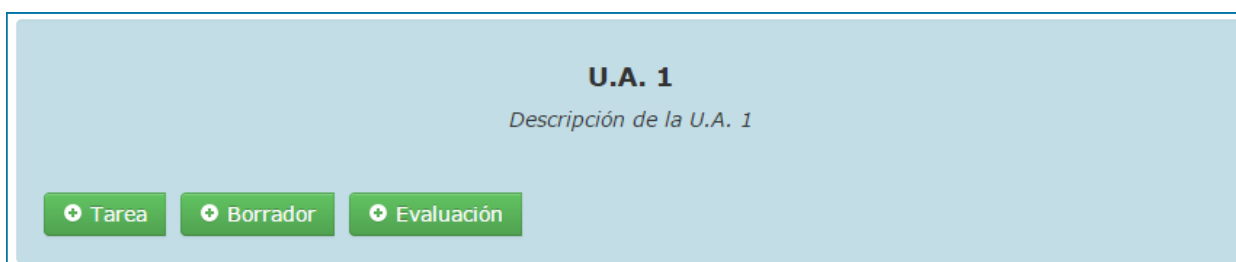
KLEARapp

Evaluación

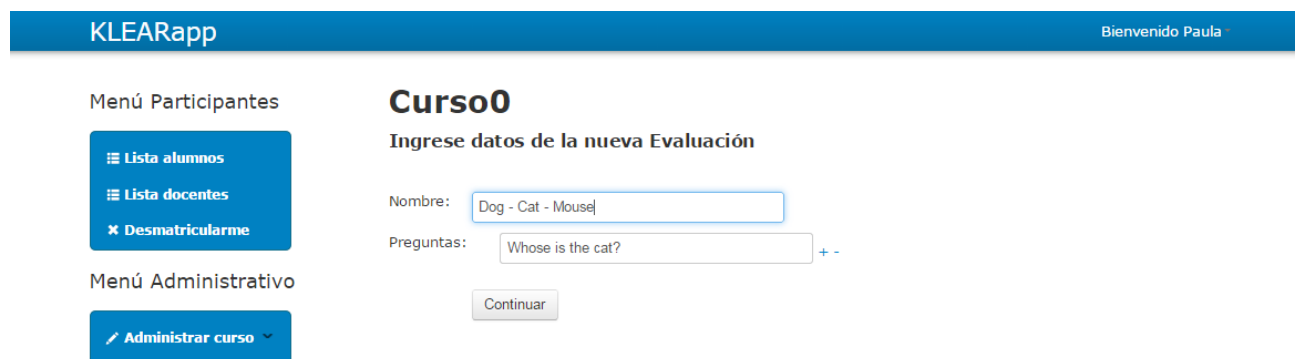
Crear Evaluación

Una vez que se crea una unidad de aprendizaje, es posible crear una evaluación, siguiendo el siguiente procedimiento.

1. Hacer click en el botón “Evaluación”



2. Se despliega un formulario para completar con los datos de la nueva evaluación. Para finalizar el proceso hacer clic en “Continuar”.



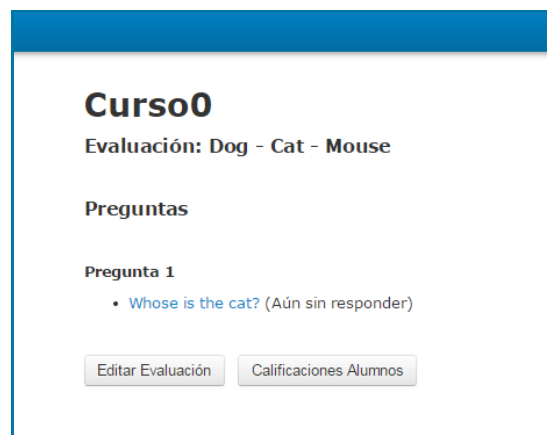
KLEARapp

Evaluación

Ingresar respuesta correcta

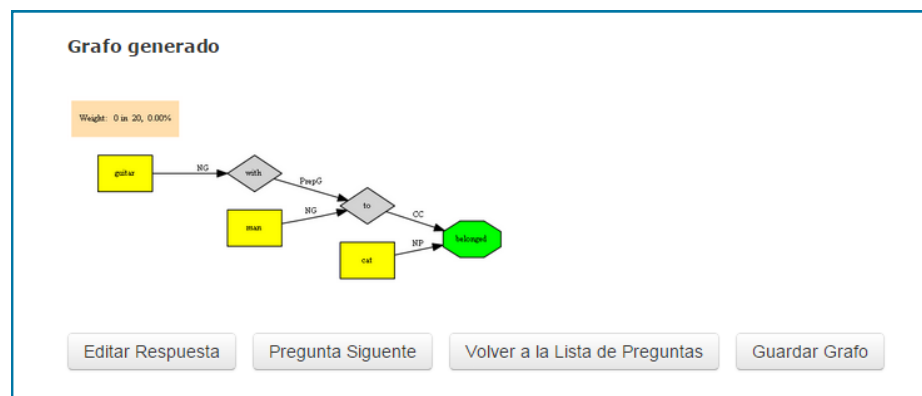
Una vez creada la evaluación y formulada las preguntas, el docente debe ingresar las respuestas correctas en base al texto referencia. Para eso se sigue el siguiente procedimiento.

1. Al ingresar a la evaluación, se despliega una lista con las preguntas creadas. El docente selecciona la pregunta.



2. Luego el docente elige aquellas sentencias que forman la respuesta correcta y hace clic en el botón “**Guardar respuesta**” para almacenarla.

La aplicación muestra el grafo de la respuesta con su puntaje correspondiente.

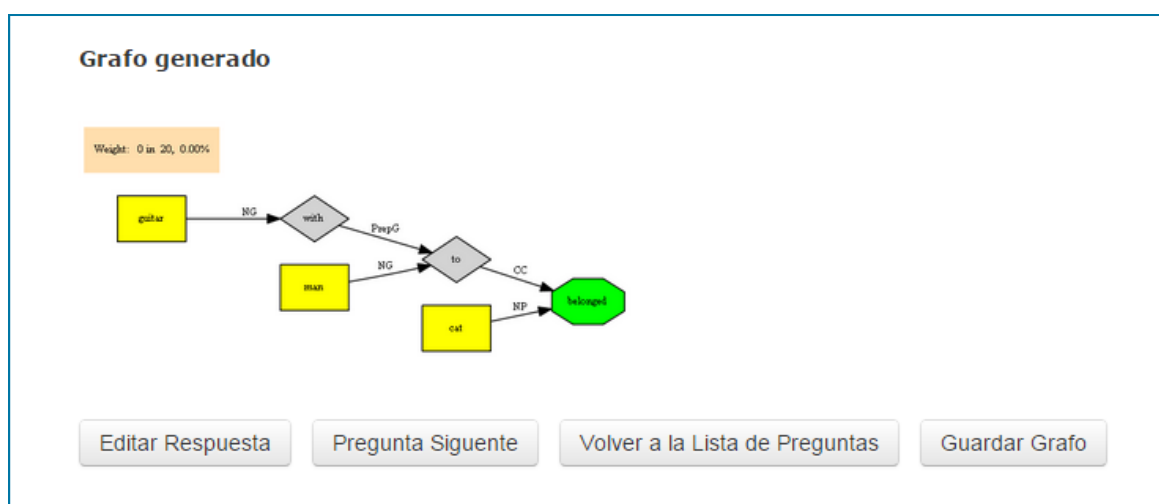


KLEARapp

Evaluación

Grafo de la respuesta correcta

Una vez armada y guardada la respuesta correcta se muestra su grafo.



Opciones para grafo generado

Editar respuesta - permite al docente volver a escribir la respuesta a la pregunta correspondiente.

Pregunta siguiente - se dirige al docente a la siguiente pregunta.

Volver a la lista de Preguntas - se vuelve a la pantalla donde se listan todas las preguntas.

Guardar Grafo - Permite guardar el grafo.