

Instituto de Computación  
Facultad de Ingeniería  
Universidad de la República

**Generador de ejercicios interactivos como herramienta de educación a distancia**

*Informe de Proyecto de Grado presentado al Tribunal Evaluador como requisito de graduación de la carrera Ingeniería en Computación.*

Montevideo - Uruguay - 2011

<i>Autores:</i>	<i>Supervisores:</i>
Nicolás MEDEIROS	Ing. Antonio López Arredondo
Guillermo PÉREZ	Dr. Alvaro Margolis Hirt
Nicolás SNIADOVER	

# Agradecimientos

Los miembros de este equipo queremos agradecer:

A nuestras familias, las que nos brindaron apoyo y confianza a lo largo de este período tan importante en nuestra carrera universitaria.

A nuestro tutor Antonio López y también a Rodrigo Beceiro, Marcos Dutto, Marcelo Canina, Dr. Álvaro Margolis y demás integrantes de la empresa Evimed que colaboraron activamente a lo largo del proyecto.

A nuestros amigos, compañeros de estudio, compañeros de trabajo y a todos aquellos que de alguna manera colaboraron y/o apoyaron durante este período.

# Resumen Ejecutivo

Con el objetivo de enriquecer la experiencia del estudiante en la realización de actividades educativas, se planteó por parte de la empresa Evimed [1] la necesidad de contar con un módulo generador de casos interactivos que permita crear un nuevo tipo de actividad en el entorno de la plataforma Moodle.

A los efectos del proyecto, se considera a los *casos interactivos* como una herramienta de evaluación, que utiliza tecnologías de la información y comunicación con el fin de enfrentar al estudiante a diferentes *escenarios* o *situaciones*. Para cada escenario se ofrece un número de opciones (*decisiones*) que el estudiante deberá elegir. En la descripción de cada escenario es posible utilizar textos y diferentes tipos de información multimedia, como vídeos, fotos o audio. Con dicha información, el estudiante deberá tomar una decisión y recibirá, si corresponde, *retroalimentación* en función de las decisiones que ha tomado. Dicha retroalimentación puede ser, a su vez, material multimedia. El sistema mostrará información sobre el promedio de decisiones que han tomado los demás cursantes, permitiendo al estudiante comparar la suya con las de sus pares. Cada decisión lleva al estudiante a un nuevo escenario, de forma que el estudiante “recorre” interactivamente el caso, pudiendo volver atrás sobre su decisión, aprendiendo en dicho proceso.

Si bien los casos interactivos son utilizados frecuentemente por distintas organizaciones, en todos los casos identificados se hace un desarrollo específico para su construcción; no se encontró una herramienta que se adapte a los diferentes casos de las organizaciones y no requiera programación extra por parte de personas especializadas.

La herramienta desarrollada en este proyecto permite definir un flujo de situaciones posibles y decisiones que el estudiante puede tomar frente a dichas situaciones. Los casos generados son independientes de la presentación (interfaz de usuario final), permitiendo al personal especializado expandir la gama de presentaciones utilizando simplemente la misma fuente de información. Esto brinda la posibilidad al docente de seleccionar diferentes tipos de presentaciones para mostrar el caso al estudiante. La herramienta genera información que sirve para análisis estadístico a nivel de estudiante

y también a nivel de cada caso. La solución planteada está completamente integrada a Moodle como un Módulo de Actividad.

### **Palabras clave**

Casos interactivos, interacción, educación, retroalimentación, e-learning, Moodle, web 2.0, aprendizaje.

# Índice General

[Agradecimientos](#)

[Resumen](#)

[Índice General](#)

[Introducción](#)

[Objetivos del proyecto](#)

[Aportes del proyecto](#)

[Organización del documento](#)

[Marco conceptual](#)

[Análisis del problema](#)

[Solución planteada](#)

[Descripción general](#)

[Implementación](#)

[Limitaciones](#)

[Dificultades encontradas](#)

[Testing](#)

[Casos de estudio](#)

[Gestión de proyecto](#)

[Descripción del equipo y tareas desempeñadas](#)

[Conclusiones](#)

[Desarrollo del proyecto](#)

[Distribución de actividades en el proyecto](#)

[Logros obtenidos](#)

[Trabajo a futuro](#)

[Referencias](#)

[Anexos](#)

[Anexo – Manual de Usuario](#)

[Anexo – Manual de Usuario](#)

[Usuario Administrador - Instalación del Módulo 'ecgmod' como Actividad en](#)

[Moodle 2.0+](#)

[Usuario Encargado de un Curso - Configuración de Casos](#)

[Agregar una instancia del Módulo 'ecgmod' como Actividad](#)

[Ajuste de propiedades del Módulo](#)

[Luego de ingresar la información general de la Actividad, se redirige automáticamente a la sección de ajuste de propiedades del Caso.](#)

[Ajuste de propiedades del Caso](#)

[Agregar, Modificar y Eliminar los Escenarios que componen el Caso](#)

[Agregar, Modificar y Eliminar las Acciones que componen un Escenario](#)

[Usuario final, participante del Curso](#)

# Capítulo 1 - Introducción

El avance de la tecnología está teniendo un impacto importante en la educación, facilitando el acceso de forma rápida y sencilla a grandes volúmenes de información. Hoy en día contar con herramientas de comunicación como Internet, telefonía, etc. se ha vuelto imprescindible en el vivir cotidiano. En particular, el uso de Internet como herramienta de capacitación es altamente aplicado, por lo que resulta natural pensar en brindar herramientas digitales que puedan ser utilizadas en capacitación vía Internet. En el marco del proyecto de grado se propone la construcción de una herramienta de generación de casos interactivos, integrada a la plataforma Moodle.

La inquietud surge por parte de la empresa EviMed, a partir de la necesidad de implementar casos interactivos similares a los presentados en el New England Journal of Medicine [1]. A continuación se puede ver una captura de pantalla como ejemplo de un caso interactivo tomado de dicha publicación, donde se presentan diferentes opciones para que el estudiante responda. En los pasos previos al que se visualiza, se presentó la información necesaria para que el estudiante pueda responder.

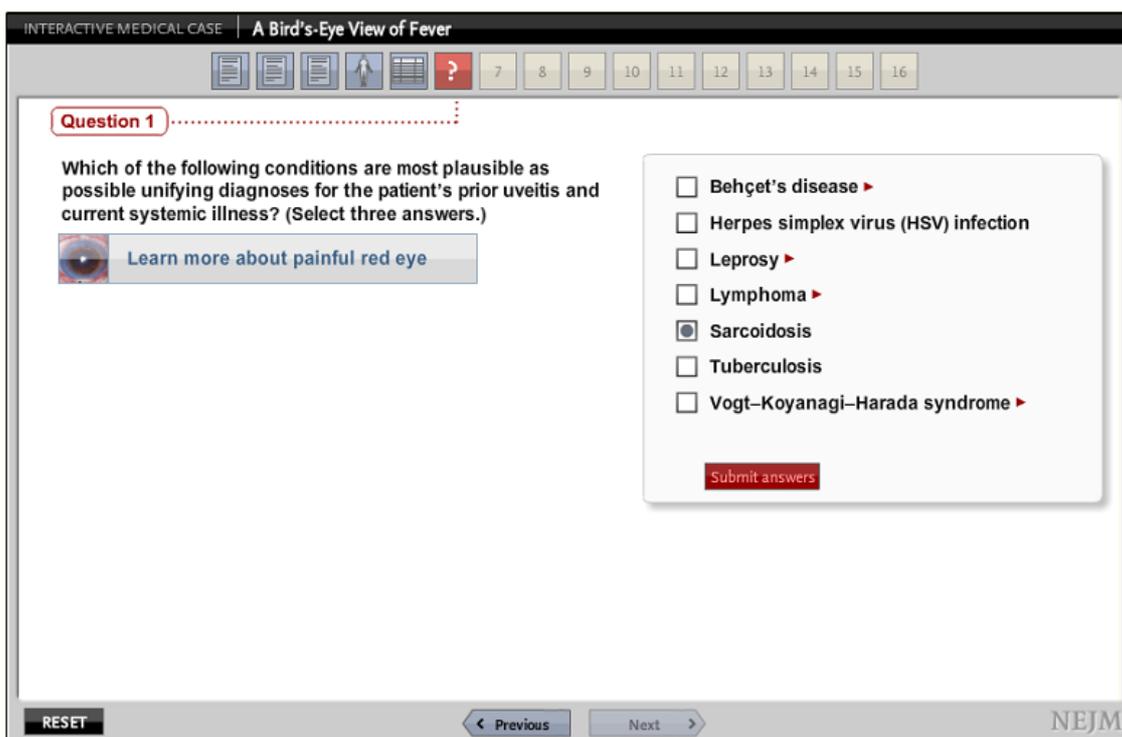


Figura 1.1 - Escenario

En la siguiente captura se puede ver un ejemplo de retroalimentación luego que el estudiante respondió en el escenario anterior. Estas características permiten al estudiante comprender por qué sus respuestas son correctas o no, recibiendo más información.

INTERACTIVE MEDICAL CASE | A Bird's-Eye View of Fever

Question 1

Which of the following conditions are most plausible as possible unifying diagnoses for the patient's prior uveitis and current systemic illness? (Select three answers.)

[Learn more about painful red eye](#)

**Feedback**

The correct answers are lymphoma, sarcoidosis, and tuberculosis.

Although it is not certain that the patient's previous eye disease is related to his current systemic illness, several disorders might explain both his past and present symptoms. Lymphoma occasionally involves the eye, mimicking uveitis that results from inflammatory conditions; it should be considered in any patient with a granulomatous disease that has systemic features.

**Sarcoidosis** has protean manifestations that include fever, sweats, and weight loss. The eye and adnexa are involved in 25 to 80% of patients.

Behçet's disease ▶  
 Herpes simplex virus (HSV) infection  
 Leprosy ▶  
 Lymphoma ▶  
 Sarcoidosis  
 Tuberculosis  
 Vogt-Koyanagi-Harada syndrome ▶

RESET Previous Next NEJM

Figura 1.2 - Feedback del escenario

Pueden verse más ejemplos de uso de casos interactivos en Educación Médica Continua en sitios como MedScape [2].

En base a los ejemplos anteriores, se realizó un análisis de las funcionalidades provistas por la versión 2.0.3 de Moodle, siendo la nueva actividad llamada *Lecciones* la que más se adaptaría a situaciones como las presentadas anteriormente. Si bien las Lecciones son un avance hacia la interactividad, aún no permiten especificar casos altamente interactivos con todas las características buscadas:

- Presentación en múltiples interfaces de usuario (incluido Flash). Esto es de gran importancia, ya que es lo que dará la flexibilidad necesaria para abarcar la mayor cantidad de casos posibles.
- Ejecución altamente desacoplada del servidor.

- Posibilidad de asignar puntaje a las decisiones tomadas.
- Visualización personalizada de la evolución del puntaje y progreso en tiempo real.
- Comparación del puntaje contra el promedio de los demás cursantes.
- Posibilidad de guardar el estado para retomar el caso en otro momento.
- Inclusión opcional de *retroalimentación (feedback)* en todas las transiciones. En la retroalimentación se muestra el promedio de respuestas de el resto de los estudiantes ante esa situación.
- Cancelación forzosa del caso si el puntaje baja de determinado umbral.
- Posibilidad de repetir el caso la cantidad de veces que se defina.
- Soporte para elementos multimedia.

Algunos de los puntos mencionados anteriormente pueden encontrarse en las *Lecciones*, pero no cumplían con las necesidades específicas del proyecto. Por ejemplo, en el caso del ítem 'asignación de puntaje', *Lecciones* cuenta con esta funcionalidad pero el concepto detrás de la calificación es diferente al manejado en este proyecto, pues la calificación es relativa al total de los puntos posibles (ej.: si se tiene un solo Escenario con dos Acciones (10 pts y 30 pts), en *Lecciones* el 100% son 40 pts, pero en este proyecto se considera 100% = 30 pts).

Es por esto que se planteó la idea de crear un Módulo generador de casos interactivos para Moodle. Los casos se crean desde Moodle como un tipo más de Actividad. La plataforma es genérica, de forma tal que puede ser utilizada por cualquier tipo de curso, independientemente del tema del mismo.

## 1.1. Objetivos del proyecto

Los objetivos se dividen en dos grandes grupos, generales y específicos. Uno de los objetivos generales en este proyecto es crear un Módulo de generación de ejercicios interactivos que este integrado a la plataforma Moodle. Otro objetivo general es realizar una publicación en algún congreso con la propuesta y el resultado del proyecto.

Dentro de los objetivos específicos se puede destacar la creación de ejercicios interactivos desde Moodle como un tipo más de Actividad. La herramienta deberá ser

general, de forma tal que pueda ser utilizada por cualquier tipo de curso independiente del tema del mismo.

Los ejercicios interactivos contarán de varios escenarios (estados) posibles y un conjunto de acciones (decisiones) que pueden ser tomadas frente a la situación planteada.

Las descripciones (tanto de las situaciones como decisiones posibles) deberán permitir cualquier tipo de contenido que pueda ser presentado en una página web (textos, imágenes, audio, videos, applets, etc).

Las transiciones tendrán asociadas 'pesos' que medirán la calidad de la decisión tomada. Dichos 'pesos' se utilizarán para la medición de la *performance* del estudiante durante la ejecución del ejercicio.

La rápida interacción es un requerimiento no funcional importante; el caso interactivo deberá poder ser ejecutado en cualquier navegador de Internet en forma totalmente interactiva y asincrónica (minimizando las conexiones con el servidor, y estableciendo conexiones en el background).

El ejercicio podrá "navegarse", mientras el alumno lo esté realizando podrá avanzar y retroceder, lo que le permitirá cambiar decisiones anteriores y mejorar su puntaje aprendiendo del ejercicio.

Se proveerá un servicio que permitirá obtener la información necesaria del ejercicio para que futuros desarrolladores puedan agregar nuevas capas de presentación.

## **1.2. Organización del documento**

Los siguientes capítulos y secciones de este documento se organizan de la siguiente manera:

El capítulo 2 presenta el marco conceptual necesario para comprender adecuadamente los temas tratados en el documento.

El capítulo 3 presenta el análisis del problema, el cual consiste en el relevamiento realizado sobre los requerimientos planteados por el cliente.

El capítulo 4 presenta la solución planteada considerando la información aportada en las secciones anteriores. Se describe en alto nivel la arquitectura de la misma; los principales componentes de la solución; se proporciona una visión de la implementación y además se mencionan las decisiones tomadas.

El capítulo 5 presenta la implementación, donde se muestran las tecnologías utilizadas, testing, casos de estudio y las dificultades encontradas.

El capítulo 6 presenta la gestión del proyecto, donde se describe al equipo y las tareas desarrolladas, la distribución de las actividades, las herramientas de gestión y desarrollo utilizadas y los logros obtenidos.

El capítulo 7 presenta los trabajos futuros.

El capítulo 8 presenta las conclusiones obtenidas al finalizar el proyecto.

Al final del documento se presenta la sección de anexos.

## Capítulo 2 - Marco conceptual

En este capítulo se presenta un resumen de los conceptos básicos, tecnologías y estándares abordados a lo largo del proyecto.

### 2.1. E-Learning

Se denomina e-learning a la educación a distancia a través de canales electrónicos (redes de comunicación, internet), utilizando aplicaciones (páginas web, foros de discusión, plataforma de formación, etc.) como soporte de los procesos de enseñanza-aprendizaje. La educación a distancia es una modalidad educativa en que los estudiantes no necesitan concurrir a clases. Una de las características más atractivas de esta modalidad de estudio es su flexibilidad de horario.



*Figura 2.1 - e-learning*

Gracias a las nuevas tecnologías los estudiantes “en línea” puede comunicarse y colaborar con sus compañeros y docentes de forma sincrónica y asincrónica, sin limitaciones espacio-temporales.

Según Bou Bauzá Guillem [3], un referente del *e-Learning*, "La interactividad supone un esfuerzo de diseño para planificar una navegación entre pantallas en las que el usuario sienta que realmente controla y maneja una aplicación". Esto es apreciable a la hora de diseñar casos interactivos, ya que se requiere de una detallada planificación y elaboración.

Por otro lado, es importante destacar la opinión de Gabriel Kaplún sobre el *e-Learning* [9]. Según Kaplún, “*el aprendizaje electrónico no existe, de igual manera que no existe el aprendizaje impreso*”. El aprendizaje es una actividad que requiere de procesos de interacción, tanto con otros seres humanos como con la naturaleza y con diversos objetos. Las computadoras simplemente constituyen herramientas para el aprendizaje. En este contexto, los módulos interactivos son otra herramienta disponible para los educadores a la hora de implementar actividades de educación a distancia.

Algunas de las ventajas a destacar del uso de casos interactivos para la educación son:

- Genera un incremento en la motivación e interés por parte del estudiante al tener una participación activa.
- Brinda retroalimentación sobre la acción tomada por el usuario.
- Permite intercalar teoría y práctica en el proceso de aprendizaje.

## **2.2. Sistema de Gestión de aprendizaje**

Un sistema de gestión de aprendizaje es un software que se emplea para administrar, distribuir y controlar las actividades de formación no presencial de una institución u organización.

Algunas de las principales funciones son:

- Gestionar usuarios.
- Gestionar recursos (materiales y actividades de formación).
- Administrar el acceso.
- Controlar y hacer seguimiento del proceso de aprendizaje.
- Realizar evaluaciones.
- Generar informes.
- Servicios de comunicación (foros, etc.)

Un sistema de gestión de aprendizaje generalmente no incluye posibilidades de crear sus propios contenidos, sino que se focaliza en gestionar contenidos creados por diferentes fuentes. El trabajo de crear los contenidos para los cursos se desarrolla

mediante un Learning Content Management System (LCMS).

## 2.3. Moodle

### 2.3.1. Definición

De acuerdo al sitio web de Moodle [13], *“Moodle es un Sistema de Gestión de Cursos de Código Abierto (Open Source Course Management System, CMS), conocido también como Sistema de Gestión del Aprendizaje (Learning Management System, LMS) o como Entorno de Aprendizaje Virtual (Virtual Learning Environment, VLE). Es una aplicación web gratuita que los educadores pueden utilizar para crear sitios de aprendizaje efectivo en línea.”*

La palabra Moodle es un acrónimo de *Module Object-Oriented Dynamic Learning Environment (Entorno Modular de Aprendizaje Dinámico Orientado a Objetos)*.

Es una plataforma educativa de distribución libre que es utilizada en el ambiente de la enseñanza virtual, ayuda a los educadores a crear comunidades de aprendizaje en línea.

Fue creado por Martin Dougiamas, la primera versión apareció el 20 de agosto del 2002. Basó su diseño en las ideas del “constructivismo en pedagogía” que afirman que el conocimiento se construye en la mente del estudiante en lugar de ser transmitido sin cambios a partir de libros o enseñanzas y en el aprendizaje colaborativo. Su arquitectura y herramientas son apropiadas para clases en línea, así como también para complementar el aprendizaje presencial. Tiene una interfaz de navegador de tecnología sencilla, ligera, y compatible.

El código está escrito en PHP bajo GNU GPL versión 3.



*Figura 2.2 - Moodle*

## **2.3.2. Administración del Sitio**

Las características de administración que ofrece Moodle son:

- Administración general del sitio por parte de los usuarios con rol de Administrador.
- Personalización del sitio utilizando "temas" que redefinen los estilos, los colores del sitio, la tipografía, la presentación, la distribución, etc.
- Pueden añadirse nuevos módulos de actividades a los ya instalados en Moodle.
- Los paquetes de idiomas permiten una localización completa de cualquier idioma. Estos paquetes pueden editarse usando un editor integrado.

## **2.3.3. Administración de usuarios**

Moodle soporta varios mecanismos de autenticación a través de módulos, que permiten una integración sencilla con los sistemas existentes.

Las características principales incluyen:

- Método estándar de alta por correo electrónico: los estudiantes pueden crear sus propias cuentas de acceso. La dirección de correo electrónico se verifica mediante confirmación.
- Método LDAP: las cuentas de acceso pueden verificarse en un servidor LDAP. El administrador puede especificar qué campos usar.
- IMAP, POP3, NNTP: las cuentas de acceso se verifican contra un servidor de correo o de noticias (news). Soporta los protocolos SSL y TLS.
- Base de datos externa: Cualquier base de datos que contenga al menos dos campos puede usarse como fuente externa de autenticación.

Cada persona necesita sólo una cuenta para todo el servidor. Por otra parte, cada cuenta puede tener diferentes roles, las cuentas con rol de administrador son las que controlan la creación de cursos y determinan los docentes, asignando usuarios a los cursos.

Seguridad: los usuarios con rol de docente pueden añadir una "clave de acceso" para sus cursos, con el fin de impedir el acceso de quienes no sean sus estudiantes. Pueden transmitir esta clave personalmente o a través del correo electrónico personal, etc. Los docentes pueden dar de baja a los estudiantes manualmente si lo desean, aunque también existe una forma automática de dar de baja a los estudiantes que permanezcan inactivos durante un determinado período de tiempo (establecido por el administrador).

Cada usuario puede especificar su propia zona horaria, y todas las fechas marcadas en Moodle se traducirán a esa zona horaria (las fechas de escritura de mensajes, de entrega de tareas, etc.). También cada usuario puede elegir el idioma que se usará en la interfaz de Moodle (Inglés, Francés, Alemán, Español, Portugués, entre otros)

### **2.3.4. Administración de cursos**

El rol docente tiene control total sobre todas las opciones de un curso. Por ejemplo, puede elegir entre varios formatos de curso tales como semanal, por temas o el formato social aunque no podrá crear nuevos usuarios (la cual es una tarea del rol administrador)

En general Moodle ofrece una serie flexible de actividades para los cursos: foros, diarios, cuestionarios, materiales, consultas, encuestas, tareas, entre otras. En la página principal del curso se pueden presentar los cambios ocurridos desde la última vez que el usuario entró en el curso, lo que ayuda a crear una sensación de comunidad.

### **2.3.5. Moodle 1.9 versus 2.0**

Moodle la versión 2.0 con respecto a la 1.9 ha sufrido una importante actualización, se le aplicaron una gran cantidad de cambios para su mejora.

Algunas de las mejoras que se han realizado:

- El acceso a las actividades pueden ser restringidas por ciertos criterios como

fechas, calificaciones obtenidas, otras actividades. Los docentes pueden especificar los criterios para definir si una actividad fue finalizada por un estudiante.

- "Site-wide groups", manejar de forma simple un grupo de estudiantes enrolados en el sitio que se puedan inscribir en una actividad.
- Repositorios, la gestión de archivos a sufrido cambios en su interfaz y funcionalidad. Ahora presenta una forma estándar para acceder al banco de archivos e incluye la posibilidad de integrarse a repositorios externos de contenido como GoogleDocs[14], Flickr[15], YouTube[16], etc
- Soporte, los módulos permiten exportar sus datos a sistemas externos, esto es útil para los estudiantes, para que puedan publicar sus trabajos en foros, etc.
- Se realizó una reprogramación, modificando las interfaces tanto para el estudiante, docente y administrador, se mejoró la base de código, refactorizando y logrando un diseño mas elegante.

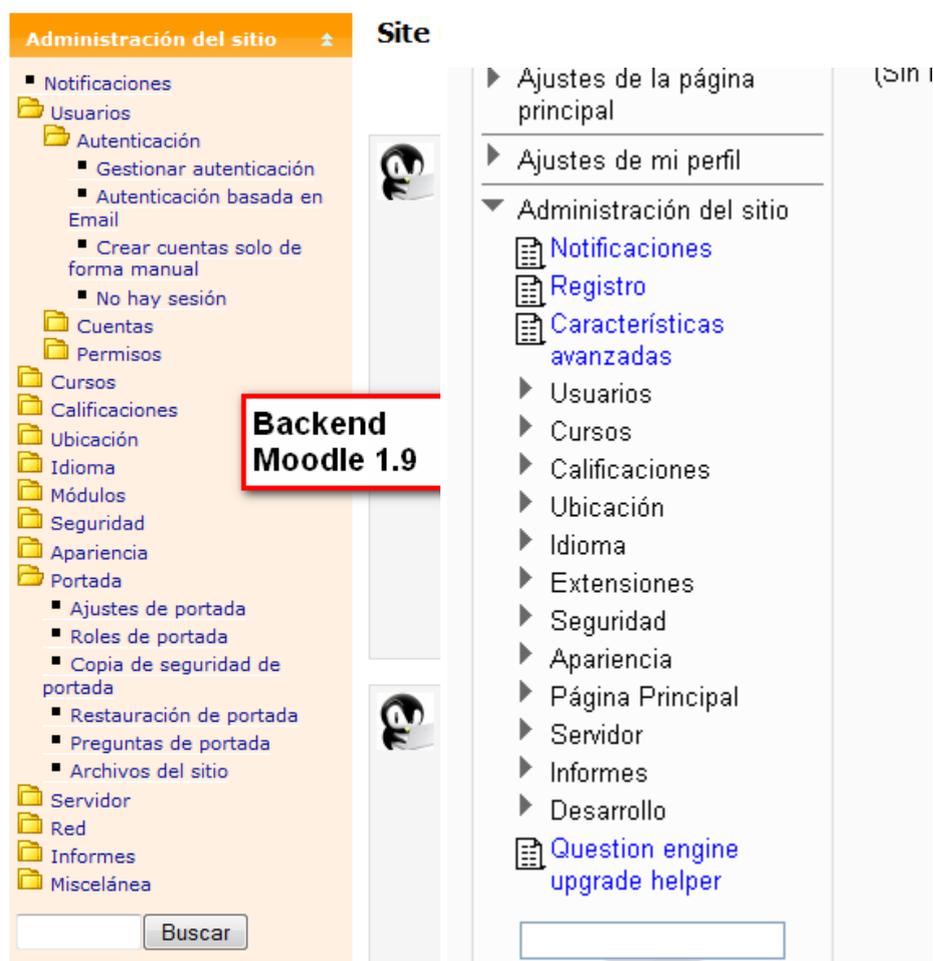


Figura 2.3 - Moodle 1.9 vs Moodle 2.0

- Actualizaciones a las últimas versiones de las herramientas de desarrollo.
  - Moodle 1.9 requiere con PHP 4.3.0, MySQL 4.1.16 o Postgres 8.0 o MSSQL 9.0 o Oracle 9.0
  - Moodle 2.0 requiere PHP 5.3.2, MySQL 5.0.25 o Postgres 8.3 o MSSQL 2005 o Oracle 10.2

El problema más notable que se ha producido en la nueva versión 2.0 es la incompatibilidad de restauración de copias de seguridad creadas con la versión 1.9.

Cabe mencionar que en comparación con versiones anteriores, la versión 2.0 es más lenta, ya que se realizó una refactorización de su código, lo que causó un aumento en la cantidad de líneas de código y por lo tanto se debe ejecutar más código para realizar una misma acción.

Para la versión 2 es notable la falta de tutoriales y manuales (tanto para alumnos, como para docentes y administradores de la plataforma), ya que la mayoría de los manuales disponibles son para la versión 1.9.

Algunas de las razones por la que se optó por la versión 2. de Moodle son:

- Es la última versión del mercado, que incluye correcciones y mejoras.
- Cambios importantes en el núcleo, que implican refactorización del código.
- Facilidad de adaptación para futuros desarrollos, debido a que permite cambios en pro de mejorar.
- Interfaz más intuitiva y amigable.

### 2.3.6. Recursos y Actividades

Dentro de los principales módulos que componen a Moodle se pueden destacar el módulo de recursos y el de actividades.

Los recursos se definen en Moodle como *“información que el profesor desea facilitar a los alumnos. Pueden ser archivos preparados y cargados en el servidor; páginas editadas directamente en Moodle, o páginas web externas que aparecerán en el curso.”*

Uno de los componentes importantes dentro del proceso de enseñanza en Moodle son las actividades, ya que permiten vincular el aprendizaje teórico con su aplicación y evaluar dicho proceso. Las actividades son elementos que involucran al estudiante al realizar un trabajo, pudiendo ser sobre un recurso que ha utilizado o no.

En el marco de éste proyecto, uno de los objetivos específicos es la creación de ejercicios interactivos desde Moodle, como un tipo más de Actividad que pueda ser utilizada por cualquier tipo de curso independiente del tema del mismo.

## Capítulo 3 - Análisis del problema

Esta sección describe el relevamiento realizado sobre los requerimientos planteados por el cliente.

Como se vió en el capítulo “Objetivos del Proyecto”, los principales requerimientos detectados para el módulo, son:

- Integración con Moodle: los casos deberán ser creados desde Moodle, como cualquier otra actividad
- Desacoplar la interfaz de forma de permitir crear nuevas interfaces de forma sencilla
- Minimizar los accesos al servidor.

A nivel de diseño, los casos interactivos son modelados como grafos, donde los nodos representan los estados posibles y las transiciones representan acciones que pueden ser tomadas frente a la situación planteada.

A continuación, se muestra un ejemplo:

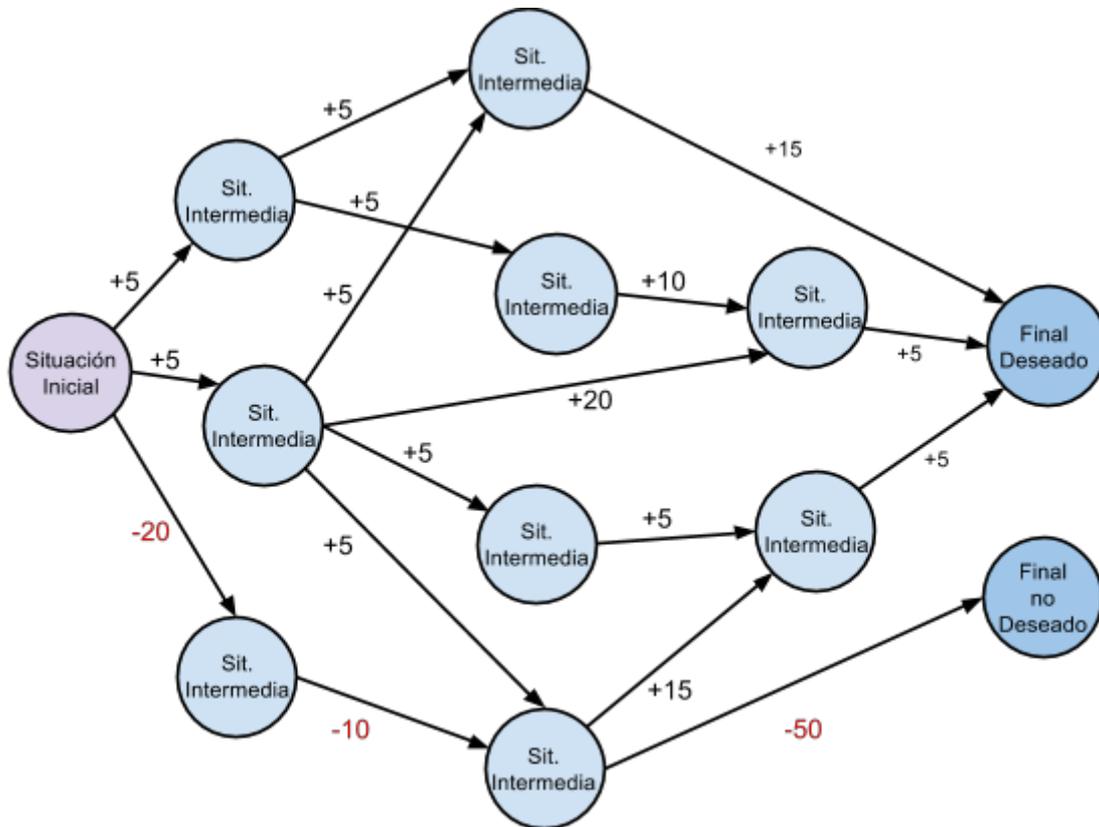
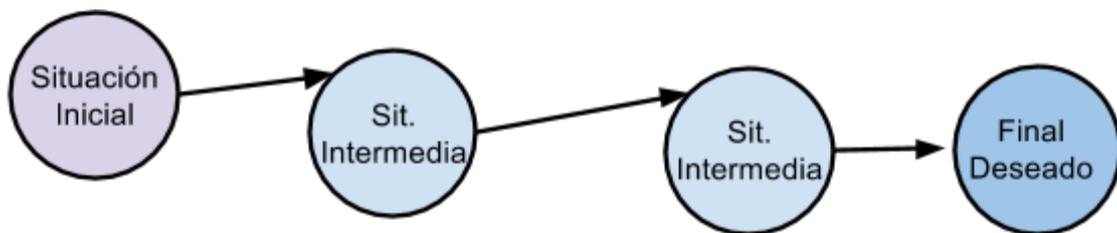


Figura 3.1 - Ejemplo de un grafo para un caso



Las descripciones (tanto de las situaciones como de las decisiones posibles) podrán contener cualquier tipo de contenido que puede ser presentado en una página web (textos, imágenes, audio, videos, applets, etc).

Las transiciones entre situaciones (consecuencia de la elección de una opción por parte del estudiante) pueden tener un peso asociado para representar la calidad de la elección y una retroalimentación (feedback) para el estudiante.

La experiencia del usuario y la rápida interacción son requerimientos importantes a tener en cuenta; el caso interactivo puede ser ejecutado en cualquier navegador de

Internet en forma totalmente interactiva y asincrónica (minimizando las conexiones con el servidor, y estableciendo conexiones asincrónicas en el background).

Una vez que el estudiante ha avanzado en el ejercicio, puede volver hacia atrás para cambiar decisiones anteriores y mejorar así su puntaje, aprendiendo del ejercicio y de la retroalimentación provista por el propio sistema.

Para cada caso definido se podrá utilizar diferentes interfaces, para ello se proveerá una API (Application Programming Interface) de forma que futuros desarrolladores puedan agregar nuevas capas de presentación.

# Capítulo 4 - Solución planteada

En esta sección se presenta la solución planteada considerando la información aportada en las secciones anteriores. Se describe en alto nivel la arquitectura de la misma y sus principales componentes, se proporciona una visión de la implementación y además se mencionan algunas de las decisiones tomadas.

## 4.1. Descripción general

La búsqueda de la solución comienza a partir del objetivo principal que es crear un módulo de generación de casos interactivos que esté integrado a la plataforma Moodle, y se funda en los dos siguientes pilares:

- que la ejecución sea fluida
- que puedan implementarse diferentes interfaces sin afectar al resto de la plataforma.

Se plantearon dos alternativas posibles buscando atacar ambos factores:

- Generación de toda la interfaz a partir del caso: esto implicaba generar código HTML, Javascript, CSS a partir del caso para luego ser “consumido” por el cliente. Esta solución permite que todo el caso se maneje como un “paquete” independiente y que por lo tanto se minimice el acceso al servidor. El caso sería cargado una única vez y toda la información estaría disponible para el cliente. La gran desventaja de este enfoque es que la implementación de nuevas interfaces implican la modificación del módulo de generación, para que se pueda generar dicha nueva interfaz
- Generación de la información del caso: en este caso se genera toda la información del caso de forma que el cliente (navegador que utiliza el estudiante) pueda “consumirla” de la manera que se desee. Esto permite minimizar los accesos al servidor, ya que se cuenta con toda la información del caso y a su vez se pueden crear nuevas interfaces que utilicen dicha información para presentar el caso de diferentes maneras.

De las opciones mencionadas anteriormente, se optó por la segunda alternativa, proponiendo una solución que consiste en dividir el problema en dos módulos independientes, uno de *configuración* y otro de *consumo del caso*, ambos integrados a Moodle.

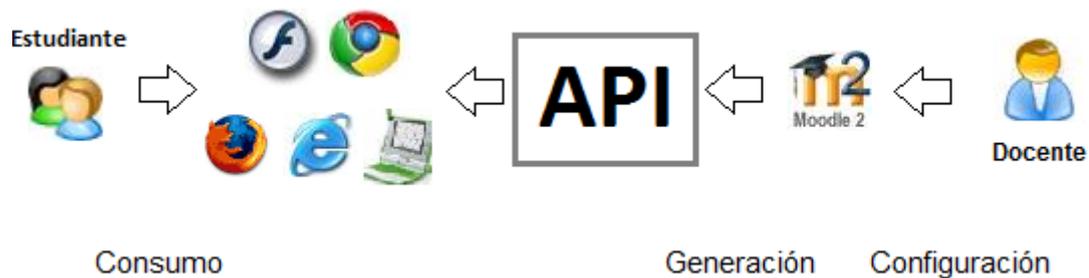


Figura 4.1 - Flujo

De esta manera se logra desacoplar ambos módulos, y con ello alcanzar uno de los principales objetivos de diseño. Aunque el módulo de *configuración* se encuentra altamente integrado a Moodle, el módulo de *consumo del caso* puede manejarse de forma aislada, permitiendo que distintas interfaces de usuario utilicen la API como intermediario entre ambos módulos y dando la posibilidad de contar con múltiples diseños para la presentación de la información de los Casos.

#### 4.1.1 Roles

Se pueden clasificar los usuarios del sistema en cuatro perfiles:

- Estudiante: usuario destinatario de los cursos, es quien utiliza los casos interactivos. El estudiante interactúa a través del navegador con el Módulo de Consumo.
- Docente: usuario encargado de configurar los casos interactivos a través de la plataforma Moodle, utilizando el Módulo de Configuración.
- Administrador: encargado de administrar el sistema. Es el usuario que instala el módulo y sus diferentes interfaces.
- Desarrollador de la interfaz: el usuario desarrollador es el encargado de diseñar e implementar nuevas interfaces para ser usadas por el módulo. Para la implementación de dichas interfaces deberá utilizar la Biblioteca de Consumo (API).

## 4.2. Módulo de Configuración

El módulo de *configuración* consiste en una interfaz sencilla mediante la cual los usuarios docentes (o bien los responsables del dictado del curso) definen paso a paso cada Caso a generar. Para ello es necesario determinar el nombre del Caso, una introducción, y demás atributos generales del mismo (periodo de vigencia, entre otros). Luego se deben ingresar los distintos escenarios que lo conforman con sus respectivas acciones posibles a tomar, con un feedback y puntaje para cada una de las decisiones, entre otros. Cuando se finaliza la configuración, habiendo establecido el orden del flujo de trabajo dependiente de las acciones que el alumno tome, el sistema toma dicha información para generar un archivo XML (uno para cada Caso). Este archivo cuenta con toda la información que luego es requerida por el módulo de *consumo del caso*.

A continuación se muestra un diagrama que ejemplifica los pasos comentados anteriormente, en donde puede apreciarse que los pasos 4, 5 y 6 son destacados con un asterisco (\*) para indicar que estos pasos pueden repetirse varias veces.

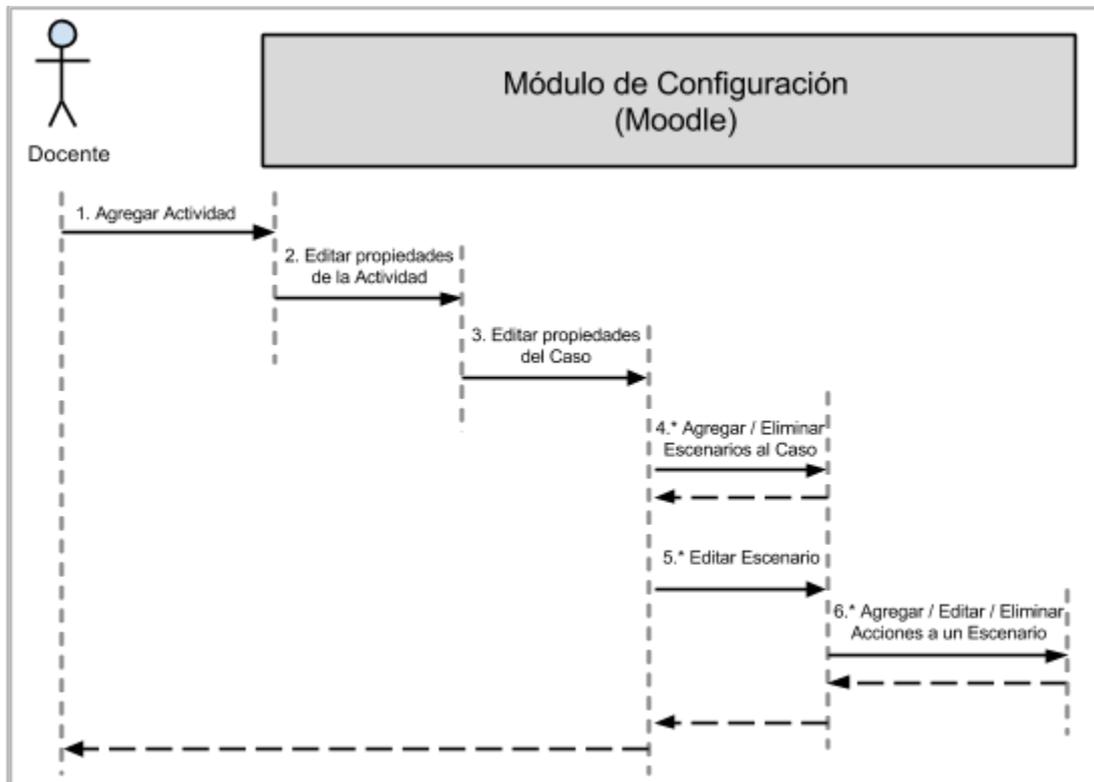


Figura 4.2 - Modulo configuración

Para el análisis del *módulo de configuración*, se realizaron algunos *mockups* (bocetos) de cómo serían las interfaces de ingreso de la información. A continuación se muestra el *mockup* del paso 3 -Editar propiedades del Caso- de la secuencia vista anteriormente.

Formulario de ingreso de un nuevo Caso. El formulario está dividido en varias secciones:

- Caso:** Campo de texto para el nombre del caso.
- Nombre Abreviado:** Campo de texto para el nombre abreviado.
- Nombre:** Campo de texto para el nombre completo.
- Progreso visible:**
- Puntaje visible:**
- Activo?:**
- Progreso Inicial:** Control deslizante con un triángulo negro.
- Puntaje Maximo:** Control deslizante con un triángulo negro.
- Descripción:** Área de texto con una barra de herramientas que incluye botones para **B** (negrita), **U** (subrayado), listas numeradas y con viñetas, un menú desplegable de **Font**, y botones de alineación y borrado.
- Pregunta Umbral:** Campo de texto.
- Puntaje Umbral:** Campo de texto con el valor **10** y botones de incremento y decremento.
- Oportunidades Rep.:** Campo de texto.
- Preguntas:** Botón para guardar o confirmar.

Figura 4.3 - Formulario de ingreso de un nuevo Caso

Siguiendo con los pasos vistos en la secuencia anterior, el *mockup* correspondiente al paso -4. Agregar / Eliminar Escenarios al Caso- sería el siguiente:

Formulario de ingreso de un nuevo Escenario (Pregunta). El formulario está dividido en varias secciones:

- Pregunta:** Campo de texto para el contenido de la pregunta.
- Nombre Abreviado:** Campo de texto para el nombre abreviado.
- Nombre:** Campo de texto para el nombre completo.
- Tipo:**
  - Inicial
  - Común
  - Final
- Descripción:** Área de texto con una barra de herramientas que incluye botones para **B** (negrita), **U** (subrayado), listas numeradas y con viñetas, un menú desplegable de **Font**, y botones de alineación y borrado.

Figura 4.4 - Formulario de ingreso de un nuevo Escenario (Pregunta)

Notar que en el *mockup* anterior la nomenclatura utilizada hace referencia a 'Pregunta' en vez de 'Escenario'. De la misma manera, en el siguiente *mockup* que se

corresponde al paso 6 -Agregar / Editar / Eliminar Acciones a un Escenarios- se hace referencia a 'Respuesta' en vez de 'Acción'. Este cambio ocurre porque luego de haber realizado los prototipos y ya comenzado el desarrollo, momento en el cual se tomó la decisión de cambiar la nomenclatura, ya que se consideró más adecuado utilizar los términos 'Escenario' y 'Acción'.

The image shows a web form for adding a new action (response). The form is enclosed in a rectangular border and contains the following elements:

- Respuesta:** A single-line text input field.
- Texto botón:** A single-line text input field.
- Descripción:** A rich text editor with a toolbar containing icons for bold (B), underline (U), bulleted list, numbered list, and indent, along with a font color selector and undo/redo buttons.
- Puntaje:** A numeric input field with the value '10' and up/down arrow buttons.
- Progreso:** A numeric input field with the value '10' and up/down arrow buttons.
- Pregunta Destino:** A single-line text input field.
- Feedback activo:** A checkbox that is currently checked.
- Feedback:** A rich text editor with a toolbar similar to the one for 'Descripción'.
- + Add:** A button located at the bottom left of the form.

Figura 4.5 - Formulario de ingreso de una nueva Acción (Respuesta)

### 4.3. Módulo de Consumo

El módulo de *consumo del caso* será un “simple” consumidor de información brindada por la API, quien funciona como nexo entre ambos módulos. Dicha API, ante la solicitud de un Caso por parte del *módulo de consumo del caso*, realiza la consulta correspondiente al *módulo de configuración*, obteniendo el XML mencionado anteriormente, del cual extraerá la información para almacenarla en memoria. Esto permite que durante la ejecución del Caso la interacción con el *módulo de*

*configuración* sea mínima, optimizándose así la experiencia del usuario, ya que se minimizan los tiempos de espera entre los diferentes escenarios. El módulo cuenta con algunas interfaces ya diseñadas que se pueden seleccionar al momento de la configuración, y además se brinda un mecanismo de 'themes' que permite agregar nuevas interfaces.

A continuación se muestra un ejemplo de interacción entre los diferentes componentes del sistema:

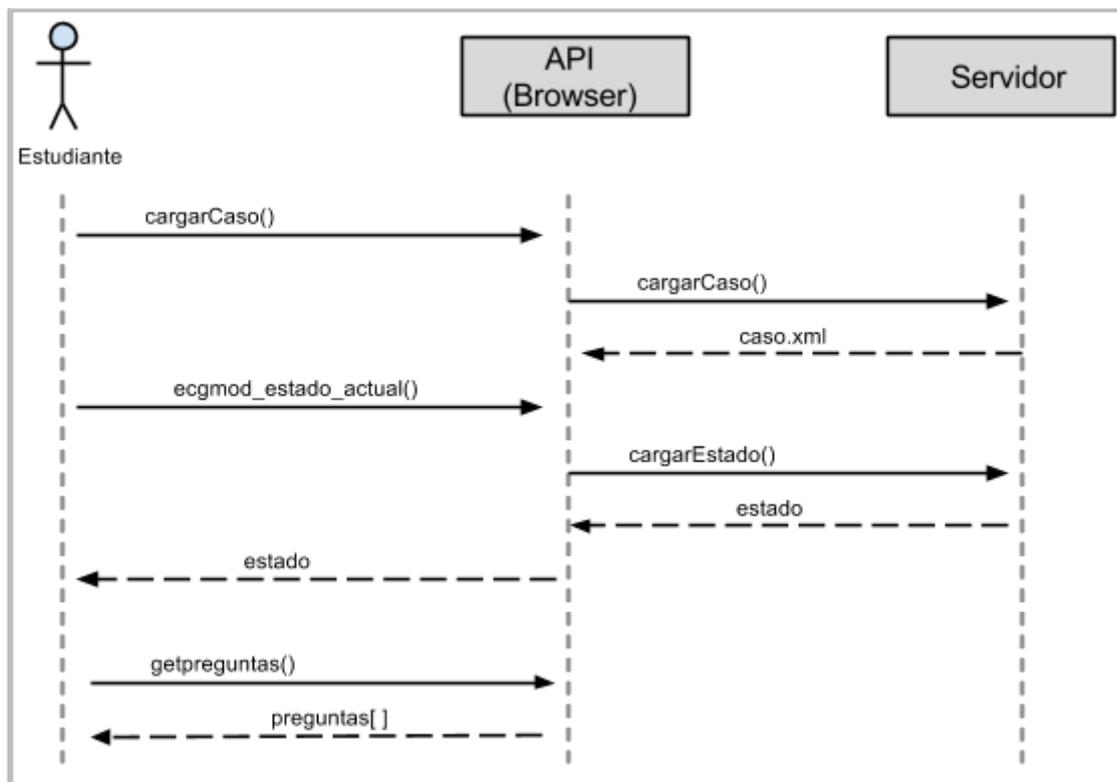


Figura 4.6 - Interacción entre componentes

Cuando se quiere obtener la información del caso, el navegador que está ejecutando el Estudiante utiliza la operación *cargarCaso()*. En ese momento la API se encarga de realizar una llamada asincrónica (utilizando AJAX) para cargar la información del caso. Una vez cargada dicha información, la interacción con el servidor se limita a las operaciones de almacenar *estado* e *historia*. Dichas operaciones son realizadas asincrónicamente con el objetivo de no bloquear la interacción del usuario con la aplicación.

### 4.3.1. Prototipos *Pestañas* y *Acordeón*

Para el análisis del *módulo de consumo*, se realizaron dos *prototipos* de cómo serían las interfaces presentadas al *estudiante*. A continuación se muestran dos figuras que corresponden a los prototipos '*Pestañas*' (tabs) y '*Acordeón*':



#### Problema de Ingeniería de Software: (Puntaje Máximo 100)

**Introducción:** La empresa NNG ha contratado a la FING para la realización de un proyecto de gran porte. El mismo está enfocado a acoplarse a un servicio de capacitación a distancia. Las características del proyecto son:

- Brindar el servicio a través de la web, funcionando únicamente on-line.
- La cantidad estimada de usuarios conectados simultáneamente es de 100.
- La interfaz debe brindar una alta performance en tiempos de respuesta.

Modelo
pregunta 2
pregunta 3

El proceso de obtención de requisitos involucra a varios actores que participan en distintas etapas del mismo.

- A. Especialista técnico, Administrador, Diseñador y Responsable de configuración.
- B. Analista, Verificador, Diseñador y Cliente.
- C. Verificador, Cliente, Responsable de calidad, Especialista técnico.

Opción A
Opción B
Opción C
<< Volver

Figura 4.7 - Prototipo de interfaz *Pestañas*

**Problema de Ingeniería de Software: (Puntaje Máximo 100)**

**Introducción:** La empresa NNG ha contratado a la FING para la realización de un proyecto de gran porte. El mismo está enfocado a acoplarse a un servicio de capacitación a distancia. Las características del proyecto son:

- Brindar el servicio a través de la web, funcionando únicamente on-line.
- La cantidad estimada de usuarios conectados simultáneamente es de 100.
- La interfaz debe brindar una alta performance en tiempos de respuesta.

▶ **1 - Modelo**

▶ **2 - Calidad**

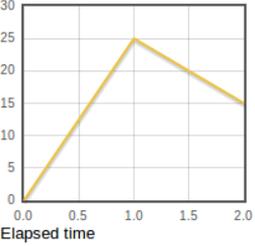
▼ **3 - Requisitos**

El proceso de obtención de requisitos involucra a varios actores que participan en distintas etapas del mismo.

A. Especialista técnico, Administrador, Diseñador y Responsable de configuración.  
 B. Analista, Verificador, Diseñador y Cliente.  
 C. Verificador, Cliente, Responsable de calidad, Especialista técnico.

**Opción A**   **Opción B**   **Opción C**

Progress

Elapsed time	Value
0.0	0
0.5	10
1.0	25
1.5	20
2.0	15

Figura 4.8 - Prototipo de interfaz Acordeón

### 4.3.2. Nuevas interfaces

La solución alcanzada cuenta con dos modelos de interfaces implementados a partir de los prototipos y además es posible integrar otras interfaces desarrolladas por terceros. Para ello debe utilizarse el mecanismo de integración de 'themes' que brinda la herramienta, el cual consiste en cumplir con dos requisitos: encapsular todos los archivos necesarios en una sola carpeta y contar con un archivo que se encargue de renderizar la interfaz. Luego simplemente resta agregar dicha carpeta a la estructura de archivos del módulo. Siguiendo estas especificaciones el docente que configure el Caso contará con dicha interfaz al momento de elegir el 'Tema de la interfaz'.

Para ver detalles de cómo integrar nuevas interfaces al módulo, dirigirse al 'Anexo 2 - Manual de usuario desarrollador', sección 'Manual para desarrollo de interfaces (themes)'.

## 4.4. Arquitectura de la solución

A continuación se muestra un diagrama en alto nivel de la arquitectura planteada para la solución, siguiendo con la explicación y justificación de los componentes utilizados en la misma y sus relaciones.

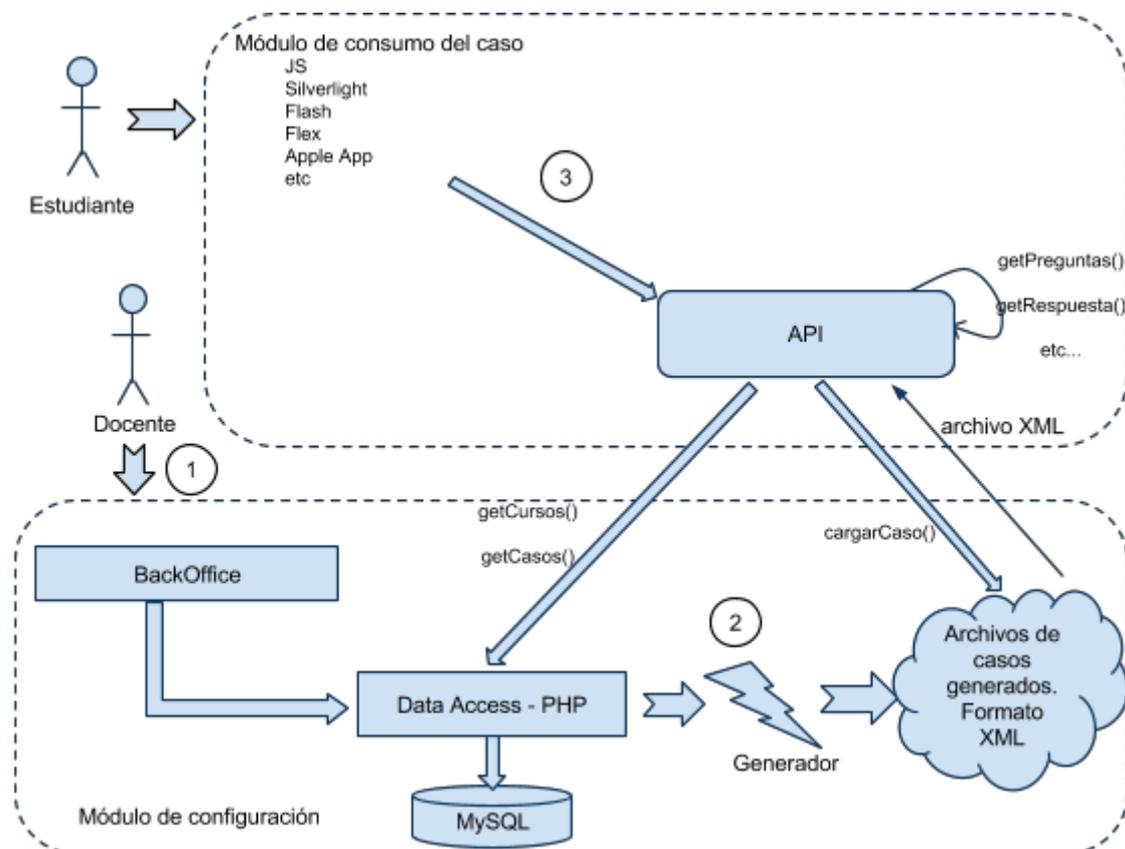


Figura 4.9 - Arquitectura

En la imagen anterior se puede ver también el orden en el cual se suceden los eventos, donde el docente primero configura un caso, luego lo genera para finalmente ser consumido por el estudiante.

#### 4.4.1 Docente

El docente interactúa con Moodle para crear y configurar nuevos casos. A continuación se muestran capturas de pantalla del proceso de creación de un nuevo caso para un curso:



Figura 4.10 - Selección de actividad

**Caso**

Nombre\* ?

Disponible desde

Disponible hasta

Tema para la interfaz

**Ajustes comunes del módulo** Ocultar Avanzadas

Modo de grupo ?

Agrupamiento\* ?

Visible

Número ID ?

En este formulario hay campos obligatorios\*

Figura 4.11 - Ajustes de la actividad

Como puede verse, el módulo está integrado a Moodle, lo cual resulta amigable al momento de ser utilizado por los docentes que ya conocen dicha plataforma.

#### 4.4.2. Interfaz de Configuración (Back-Office)

Es uno de los componentes más importantes de la solución desde el punto de vista del docente, principalmente porque es en donde cada uno de los Casos, Escenarios, etc. son definidos y configurados. Este componente será utilizado por docentes (o quien corresponda) para dar inicio al uso del sistema.

Tal como se muestra en el diagrama anterior, el docente es quien interactúa con el *Back-Office* o *Módulo de Consumo*. El detalle de esta interacción se encuentra en la sección 4.2. Módulo de Consumo.

### 4.4.3. Data-Access PHP

Este componente se encarga del acceso a la base de datos del sistema, aislando los componentes de *Back-Office* y *API* de la misma. Para el proyecto se utilizó una base de datos MySQL, sin embargo dicha tecnología podría ser cambiada por otra (por ejemplo PostgreSQL), siendo el cambio transparente a los demás componentes.

El componente de data-access es brindado por la propia plataforma Moodle, siendo sus principales ventajas:

- Abstracción de la tecnología utilizada para las bases de datos, como se vió anteriormente.
- Filtrado de información ingresada por el usuario para prevenir problemas de seguridad (por ejemplo, inyecciones sql).
- Brinda funciones para operaciones comunes, por ejemplo, obtener registro por identificador, obtener todos los registros, etc.

### 4.4.4. Generador – Archivos XML

Uno de los requerimientos no funcionales más importantes que debe brindar la aplicación es la rápida interacción con el usuario. La solución planteada propone el uso de un generador de archivos XML. Cada uno de estos archivos contiene un Caso con su respectiva información (Escenarios, Acciones, etc.) definidos en la etapa de *Back-Office*. Esta solución permite que la API genere rápidamente una respuesta a quien “la está consumiendo”, cargando en memoria la información necesaria para realizar las transiciones y la mayoría de las interacciones correspondientes al Caso.

Hay que tener en cuenta que esta forma de minimizar las conexiones con el servidor para brindar la *performance* adecuada puede significar una demora inicial al momento

de cargarse cada Caso. Es posible minimizar el impacto de la carga inicial utilizando técnicas como la opción disponible en servidores Apache de retornar los datos comprimidos.

De todas formas, ciertas operaciones, como actualizar el puntaje del usuario o guardar su estado, requieren acceder al servidor. Dichos accesos se realizan utilizando llamados AJAX, de forma de no bloquear la ejecución.

#### **4.4.5. Biblioteca para Módulo de Consumo (API)**

Esta biblioteca permite separar la capa de presentación de la capa de implementación. Ofrece al desarrollador operaciones que le permiten obtener la información correspondiente al Caso que se encuentra en el archivo XML generado por el Módulo de Configuración y además permite interactuar con el Servidor mediante algunas operaciones específicas, tales como actualización de puntaje, obtener estado del usuario, etc.

Esta interfaz de programación brinda varias ventajas para la actualización y futuros desarrollos del sistema. Algunas de las ventajas que presenta son:

- Los desarrolladores de interfaces para usuarios, pueden obtener mediante dicha biblioteca toda la información necesaria para implementar las transiciones e interacciones que los Cursos requieran.
- Los desarrolladores de Back-Office y Data-Access, simplemente dan la respuesta adecuada a las funciones que utiliza la API, sin importar cómo se resuelven dichas solicitudes.
- La base de datos, el formato y/o la forma de obtener la información pueden ser de manera transparente al usuario que está “consumiendo” la API.

#### **4.4.6. Estudiante**

La interfaz de usuario podrá constar de diferentes opciones de diseño. Particularmente la solución desarrollada cuenta con dos variantes utilizando jQuery/CSS (Tabs y Acordeón), y un prototipo para validación utilizando Flash. Este último, se pensó como

un prototipo técnico, utilizado para comprobar la viabilidad del uso de la API desde Flash. Se pudo validar la correcta interacción entre el componente Flash y la API, cargando un caso y mostrando todos sus escenarios y acciones en una película Flash.

Las variantes Tabs y Acordeón utilizan ambas la API de manera análoga, variando algunos aspectos relacionados con la diferencia que surge del diseño de las interfaces.

## 4.5. Decisiones tomadas

En esta sección se presentan las decisiones de arquitectura y diseño tomadas durante las primeras etapas del proyecto, principalmente durante la etapa de prototipación.

### 4.5.1. Decisiones de Seguridad

#### 4.5.1.1. Usuarios

Decisión tomada	El manejo de usuarios, permisos y cursos se realizan a través de Moodle haciendo uso de los roles, grupos, etc.
Situación planteada	Definir la seguridad respecto a los usuarios.
Alternativas	<ol style="list-style-type: none"> <li>1. Utilizar las funcionalidades que provee Moodle.</li> <li>2. Implementar "a medida" la seguridad necesaria.</li> </ol>
Justificación de la decisión	Se investigó sobre la implementación y funcionalidades brindadas por Moodle pudiendo determinar una baja dificultad en el aprovechamiento de las mismas. De lo contrario, debía realizarse un esfuerzo importante en desarrollar un módulo de seguridad propio.

#### 4.5.1.2. Invitados

Decisión tomada	Quienes utilicen el acceso para Invitados no podrán ejecutar los Casos, aún cuando el Curso acepte Invitados.
Situación planteada	Permitir o no, ejecutar los Casos por Invitados.
Alternativas	<ol style="list-style-type: none"> <li>1. Permitir ejecutar los Casos por Invitados.</li> <li>2. No permitir ejecutar los Casos por Invitados.</li> </ol>
Justificación de la decisión	Por razones tales como: la solución alcanzada permite determinar una cantidad variada de veces que el usuario

	<p>puede ejecutar el Caso planteado (hasta 99 veces); se registran datos de cada ejecución para obtener estadísticas; se decidió permitir la ejecución de los Casos únicamente a usuarios registrados.</p>
--	--

## 4.5.2. Decisiones de Arquitectura

### 4.5.2.1. Integración con Moodle

Decisión tomada	Implementación del back-office integrado a Moodle.
Situación planteada	Acoplamiento a la plataforma Moodle, ¿que tan integrado estará el back-office?
Alternativas	<ol style="list-style-type: none"> <li>1. Altamente integrado a Moodle.</li> <li>2. No integrado a Moodle.</li> </ol>
Justificación de la decisión	<p>Inicialmente se planteó la posibilidad de implementar un back-office por fuera de Moodle que luego permitiera una fácil integración con la plataforma, pero luego de haber estudiado la plataforma y vista la dificultad de la misma, dado que Moodle es uno de los objetivos primarios, se decidió que era mejor iniciar el desarrollo directamente en Moodle.</p>

### 4.5.2.2. Generación del XML

Decisión tomada	Al realizar la Vista Previa se genera el XML que contiene la información completa del Caso.
Situación planteada	¿Cuál es el mejor momento para generar el XML con la información del Caso?
Alternativas	<ol style="list-style-type: none"> <li>1. Generar el XML mediante un botón.</li> <li>2. Generar el XML en cada confirmación de cambios.</li> <li>3. Generar el XML al realizar la Vista Previa.</li> </ol>
Justificación de la decisión	<p>Dadas las alternativas planteadas:</p> <ul style="list-style-type: none"> <li>- se descartó la primer alternativa porque se estaría involucrando directamente al usuario <i>docente</i> requiriendo de él una acción totalmente voluntaria.</li> <li>- se descartó la segunda alternativa porque podría resultar en una sobrecarga innecesaria al servidor.</li> <li>- se optó por la tercer alternativa porque aunque el <i>docente</i> debe realizar una acción, dicha acción resulta más intuitiva e incluso “casi obligatoria” antes de publicar el Caso.</li> </ul>

#### 4.5.2.3. Formatos de intercambio

Decisión tomada	Utilizar JSON como formato para intercambio de datos entre la API y el cliente.
Situación planteada	¿Cuál es el mejor formato para intercambio de datos entre la API y el cliente?
Alternativas	1. XML 2. JSON
Justificación de la decisión	Se optó por utilizar JSON por ser liviano, sencillo de procesar y principalmente porque puede ser evaluado nativamente en JavaScript, convirtiéndolo en objetos nativos JavaScript, para su posterior uso.

#### 4.5.2.4. Versión de Moodle

Decisión tomada	Utilizar la versión 2.0.3 de Moodle en favor de la 1.9.
Situación planteada	Utilizar la versión 1.9 ya estable o apostar a las mejoras introducidas en la versión 2.
Alternativas	1. Moodle 1.9 2. Moodle 2
Justificación de la decisión	Esta decisión se tomó luego de estudiar el código y funcionalidades de ambas versiones. <a href="#">Ver comparativa en sección 2.3.5</a>

### 4.5.3. Decisiones de Diseño

#### 4.5.3.1. Nomenclatura

Decisión tomada	Cambiar la nomenclatura de 'Preguntas' a 'Escenarios' y 'Respuestas' a 'Acciones'.
Situación planteada	Modificar algunos términos utilizados hasta el momento.
Alternativas	1. Preguntas y Respuestas 2. Escenarios y Acciones
Justificación de la decisión	Luego de haber realizado los prototipos y ya comenzado el desarrollo, se consideró más adecuado utilizar los términos 'Escenario' y 'Acción', momento en el cual se tomó la decisión de cambiar la nomenclatura.

#### 4.5.3.2. Back-Office

Decisión tomada	Interfaz de Back-Office siguiendo las recomendaciones del 'Manual de Estilo de Código' que brinda Moodle.
Situación planteada	Cómo Implementar la interfaz de Back-Office.
Alternativas	<ol style="list-style-type: none"> <li>1. "Libremente" utilizando elementos y bibliotecas de interfaz propias.</li> <li>2. Seguir los lineamientos que plantea el 'Manual de Estilo de Código' de Moodle.</li> </ol>
Justificación de la decisión	Dado que uno de los objetivos del proyecto era desarrollar un módulo para la plataforma Moodle, previendo la posibilidad de presentación del módulo como posible extensión de la plataforma, se consideró apropiado realizar un esfuerzo de aprendizaje sobre el funcionamiento interno de la herramienta, la nomenclatura, recomendaciones, etc., logrando así una implementación acorde a las recomendaciones de Moodle, tanto a nivel de interfaz como de código.

#### 4.5.4. Decisiones de Implementación

##### 4.5.4.1. Prototipo utilizando Flash

Decisión tomada	Implementar un prototipo de interfaz Flash con el objetivo de comprobar el uso de la API en contextos diferentes a HTML/JS, no necesariamente con el fin de utilizar todas las funcionalidades posibles, sino que meramente para ilustrar el consumo de los servicios brindados por la API
Situación planteada	Contar con distintas interfaces de usuario, buscando un acercamiento a las plataformas RIA (Rich Internet Application) con mayor penetración del mercado tales como Java (JavaScript) y Flash.
Alternativas	Distintas opciones de plataformas RIA.
Justificación de la decisión	Se consideraron aspectos tales como esfuerzo de aprendizaje, acceso a la plataforma de desarrollo y experiencia de desarrollo de los integrantes. Las razones que motivaron esta decisión fueron que al menos uno de los integrantes contaba con conocimientos sobre HTML/JS y ninguno tenía experiencia en el uso de Flash (considerando además que no es una plataforma Open Source). Por lo tanto se decidió implementar las dos interfaces completamente en HTML/JS/AJAX. y un prototipo en Flash.

#### 4.5.4.2. Integración a Moodle

Decisión tomada	Alta integración a Moodle. Se decidió que el módulo sería un módulo Moodle, utilizando las herramientas que brinda dicha plataforma.
Situación planteada	A pesar que el módulo fue desarrollado de forma de ser sencillamente extraíble de Moodle para ser usado en otra plataforma o independientemente, debido al uso de muchas de las herramientas brindadas por dicha plataforma, existe un alto acoplamiento entre ambos sistemas.
Alternativas	Utilizar otras plataformas diferentes a Moodle. Implementar un sistema independiente, no integrado a ninguna plataforma.
Justificación de la decisión	Moodle es el ELE opensource más utilizado, lo que permite acceder a la mayor cantidad posible de potenciales usuarios.

#### 4.5.4. Decisiones de Testing

##### 4.5.4.1. Selenium como herramienta para automatizar pruebas

Decisión tomada	Utilizar la extensión Selenium IDE para Firefox como herramienta para realizar testing automatizado.
Situación planteada	Contar con alguna herramienta que permitiera agilizar la puesta a punto de la solución alcanzada.
Alternativas	Utilizar o no selenium.
Justificación de la decisión	Se puso énfasis en realizar pruebas de regresión para realizar la puesta a punto, para lo cual se consideró Selenium IDE como la herramienta más adecuada, cuyo entorno para la administración de scripts de testeo permite grabar, editar, depurar y ejecutar los tests.

##### 4.5.4.1. Resultados de ejecuciones

Decisión tomada	No registrar los resultados de las ejecuciones realizadas en testing de regresión.
Situación planteada	Registrar o no los resultados de las ejecuciones realizadas en testing de regresión.
Alternativas	1. Registrar

	2. No registrar
Justificación de la decisión	Debido a la agilidad de los lenguajes de programación utilizados (lenguajes dinámicos no compilados) y de acuerdo a la planificación realizada, muchas funcionalidades serían desarrolladas y testeadas inmediatamente realizando testing <i>manual</i> . Por lo tanto se resolvió que el <i>testing de regresión automatizado</i> se realizaría para agilizar la puesta a punto, en el último período de implementación.

## Capítulo 5 - Implementación

Esta sección proporciona una visión de la implementación de la solución planteada en el capítulo anterior. Se mencionan las tecnologías utilizadas para el desarrollo, las limitaciones del producto realizado, las principales dificultades encontradas y se hace una breve reseña sobre el *testing* realizado así como los casos de estudio implementados.

Recordando los componentes principales del sistema:

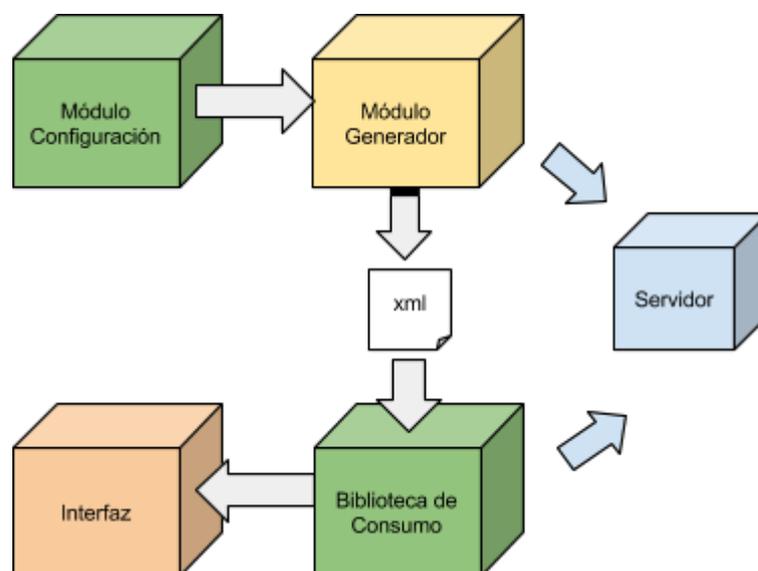


Figura 5.1 -Componentes

A continuación se describen los detalles para cada uno de los módulos.

### 5.1. Módulo de Configuración

El módulo de configuración es el encargado de brindar las funcionalidades para la configuración de los casos desde el administrador de Moodle.

La configuración de cada caso se realiza dentro de Moodle, siguiendo los mismos estándares que el resto de los módulos, por lo que resultará muy intuitivo para los

administradores que ya trabajen con la plataforma (ver Anexo – Manual de Usuario).

Para el desarrollo del módulo en Moodle se siguió un esquema predefinido por la comunidad, como por ejemplo guardando todos los archivos en una carpeta con el nombre del módulo (dentro de la carpeta *moodle/mod*). En el anexo Manual de Desarrollo de Módulos para Moodle se explican los detalles del desarrollo de módulos en Moodle y de los archivos requeridos.

A continuación se muestra un esquema en alto nivel de los diferentes componentes del módulo y sus responsabilidades:

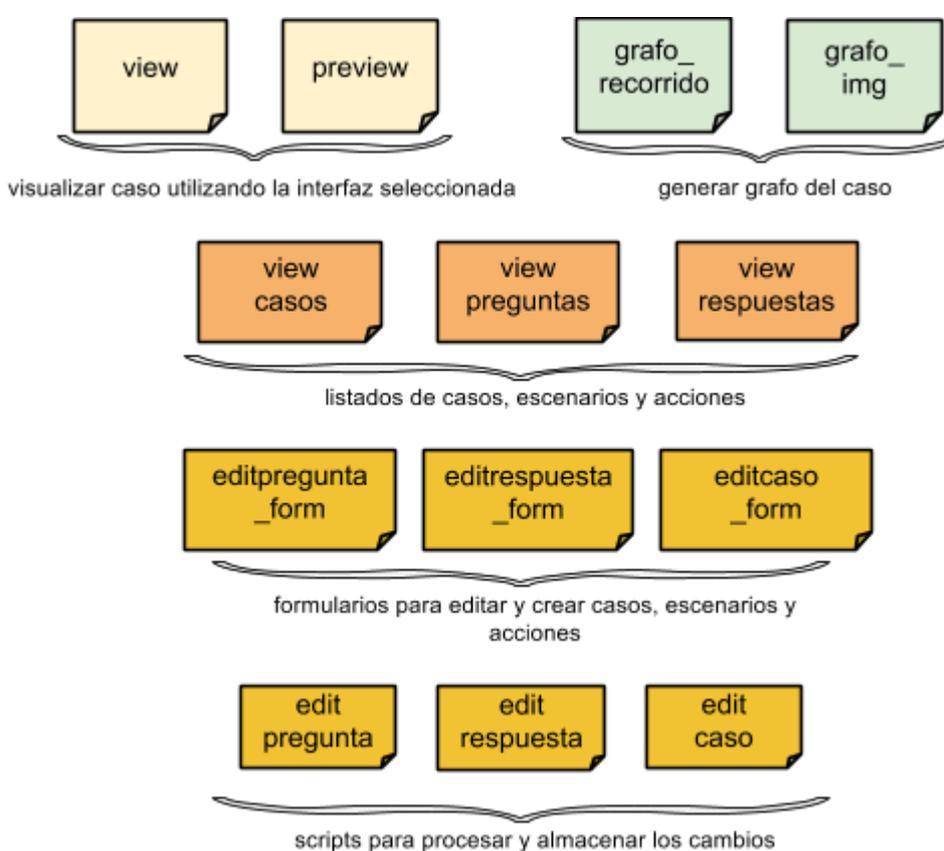


Figura 5.2 - Esquema alto nivel

Los archivos *view* y *preview* son los encargados de cargar los archivos del *theme* activo para mostrar el caso. El primero es utilizado para mostrar el caso a los estudiantes y cuenta con todas las funcionalidades; el segundo se utiliza para que el docente pueda previsualizar cómo va quedando el caso, pero no cuenta con algunas de las funcionalidades que sólo aplican en la ejecución real del caso (por ejemplo, no guarda historial ni estado, no muestra porcentajes de respuestas, etc).

*Grafo\_recorrido* y *grafo\_img* son los scripts encargados de producir una imagen con la representación gráfica del caso. El primer script genera un grafo del caso con las aristas recorridas por el usuario al realizar el caso, mientras que el segundo se utiliza para que el docente pueda visualizar el grafo del caso que está configurando.

Los archivos *view\_\** son los encargados de listar los diferentes registros: casos, escenarios y acciones.

## 5.2. Módulo Generador

El módulo generador es el componente a cargo de la generación de la información necesaria para ser utilizada por la API. Es invocado por el módulo de configuración cuando se deben generar los datos, usualmente luego de crear o actualizar un caso.



Figura 5.3 - Módulo Generador

El script *generar caso.php* toma la información de un caso específico y genera un archivo XML con toda la información. Dichos archivos son almacenados en la carpeta *moodle/xml*, cada archivo se nombra siguiendo el siguiente formato:

*caso\_idcurso\_idcaso.xml*

De esta forma, la API puede cargar los archivos de manera sencilla, concatenando el identificador del curso y el identificador del caso y buscándolo en la carpeta especificada.

La estructura del archivo XML del caso se describe a continuación:

```
<casos>
  <caso>
    <casocursoid> </casocursoid>
    <casoid> </casoid>
    <casotitulo> </casotitulo>
```

```

<titulocorto> </titulocorto>
<mostrar_progreso> </mostrar_progreso>
<progreso_inicial> </progreso_inicial>
<mostrar_puntaje> </mostrar_puntaje>
<puntaje_inicial> </puntaje_inicial>
<activo> </activo>
<descripcion> </descripcion>
<id_pregunta_umbral> </id_pregunta_umbral>
<puntaje_umbral> </puntaje_umbral>
<repeteciones_permitidas> </repeteciones_permitidas>
<vigencia_fecha_inicio> </vigencia_fecha_inicio>
<vigencia_fecha_fin> </vigencia_fecha_fin>
</caso>
<pregunta>
  <preguntacurso> </preguntacurso>
  <preguntacaso> </preguntacaso>
  <preguntaid> </preguntaid>
  <preguntatitulocorto> </preguntatitulocorto>
  <preguntatitulolargo> </preguntatitulolargo>
  <preguntatipo> </preguntatipo>
  <preguntadescripcion> </preguntadescripcion>
</pregunta>
<respuesta>
  <respuestapreguntaid> </respuestapreguntaid>
  <respuestaid> </respuestaid>
  <respuestanombre> </respuestanombre>
  <respuestaboton> </respuestaboton>
  <respuestapuntos> </respuestapuntos>
  <respuestaprogreso> </respuestaprogreso>
  <respuestadestino> </respuestadestino>
  <respuestafeedback> </respuestafeedback>
  <respuestafeedbacktexto> </respuestafeedbacktexto>
  <respuestapreguntaid> </respuestapreguntaid>
  <respuestaid> </respuestaid>
  <respuestanombre> </respuestanombre>
  <respuestaboton> </respuestaboton>
  <respuestapuntos> </respuestapuntos>
  <respuestaprogreso> </respuestaprogreso>
  <respuestadestino> </respuestadestino>
  <respuestafeedback> </respuestafeedback>
  <respuestafeedbacktexto> </respuestafeedbacktexto>
</respuesta>
</casos>

```

Los tags *pregunta* y *respuesta* se repiten para cada *escenario* (pregunta) y cada *acción* (respuesta), mientras que el tag *caso* tiene toda la información para el caso en

particular.

### 5.3. Biblioteca de Consumo

Una API es una especificación que los programas pueden seguir para poder comunicar adecuadamente distintos componentes. Es una interfaz, que define los mecanismos a través de los cuales una aplicación puede hacer uso de otra pieza de software.

Se definió una serie de operaciones y tipos de datos que son necesarios para poder mostrar y manipular los casos interactivos. Dichas operaciones están implementadas utilizando JavaScript y quedan disponibles luego de importar el archivo */moodle/mod/ecgmod/api/ecgmod\_api.js*. Ver Biblioteca de consumo (API) en el manual de usuario desarrollador.

La API permite cargar un caso, obtener todos sus escenarios, las acciones en cada escenario, el estado del usuario, guardar historial y demás operaciones necesarias para implementar un caso interactivo.

Las funciones de la API que retornan datos complejos lo hacen retornando dichos datos en formato JSON (JavaScript Object Notation). JSON es un formato liviano para intercambio de información, al igual que XML, pero con la ventaja de ser más liviano y sencillo de procesar y que además puede ser evaluado nativamente en JavaScript, convirtiéndolo en objetos nativos JavaScript, para su posterior uso.

La API puede ser consumida por diferentes tipos de aplicaciones. Por ejemplo, se puede implementar una interfaz Flash, la cual puede invocar a las funciones de la API y luego procesar los datos retornados, ya que es posible manipular datos JSON desde Flash. Para validar este caso se implementó un prototipo en Flash, que consume los servicios de la API y manipula los objetos JSON retornados.

### 5.4. Modelo de datos

En esta sección se destacan algunos puntos importantes sobre la relación existente

entre las entidades de Moodle y las nuevas entidades que conforman el nuevo módulo y además se muestra un diagrama del modelo de datos utilizado en la solución lograda.

Algunas de las entidades de Moodle que necesariamente “entran en juego” al momento de utilizar la plataforma son Usuarios (*mdl\_user*) y Cursos (*mdl\_course*). Como se verá más adelante en esta sección, algunas de las entidades de la solución alcanzada hacen referencia a registros existentes en estas tablas para mantener una relación entre los Casos y los Usuarios de los Cursos.

La primer y única entidad que Moodle crea en su base de datos al momento de agregar un nuevo módulo lleva como nombre *prefijomoodle\_nombredelmódulo* (*mdl\_ecgmod*), la que mantiene información correspondiente a cada instancia de una nueva *actividad* del *módulo* y la relaciona con el *curso* correspondiente. En la solución alcanzada se agregaron además algunos *campos* a dicha tabla tales como fechas de vigencia, tema de la interfaz a utilizar, entre otros.

Las demás tablas que componen la solución alcanzada y se muestran en el siguiente diagrama llevan como nombre *prefijodelmódulo\_nombredeentidad* (por ejemplo, *mdl\_ecgmod\_casos*) y son las entidades que mantienen la información correspondiente a cada Caso. Las relaciones entre las tablas *mdl\_ecgmod\_casos*, *mdl\_ecgmod\_preguntas* (escenarios) y *mdl\_ecgmod\_respuestas* (acciones) son 1:N, ya que cada Caso está compuesto por dos o más Escenarios y cada uno de éstos por al menos una Acción. Las tablas *mdl\_ecgmod\_estado* y *mdl\_ecgmod\_historia* mantienen información de la situación actual de cada usuario y del historial de ejecuciones respectivamente. La información guardada en estas tablas permite brindar funcionalidades tales como ‘abandonar y continuar/recomenzar’, obtener datos estadísticos, entre otros..

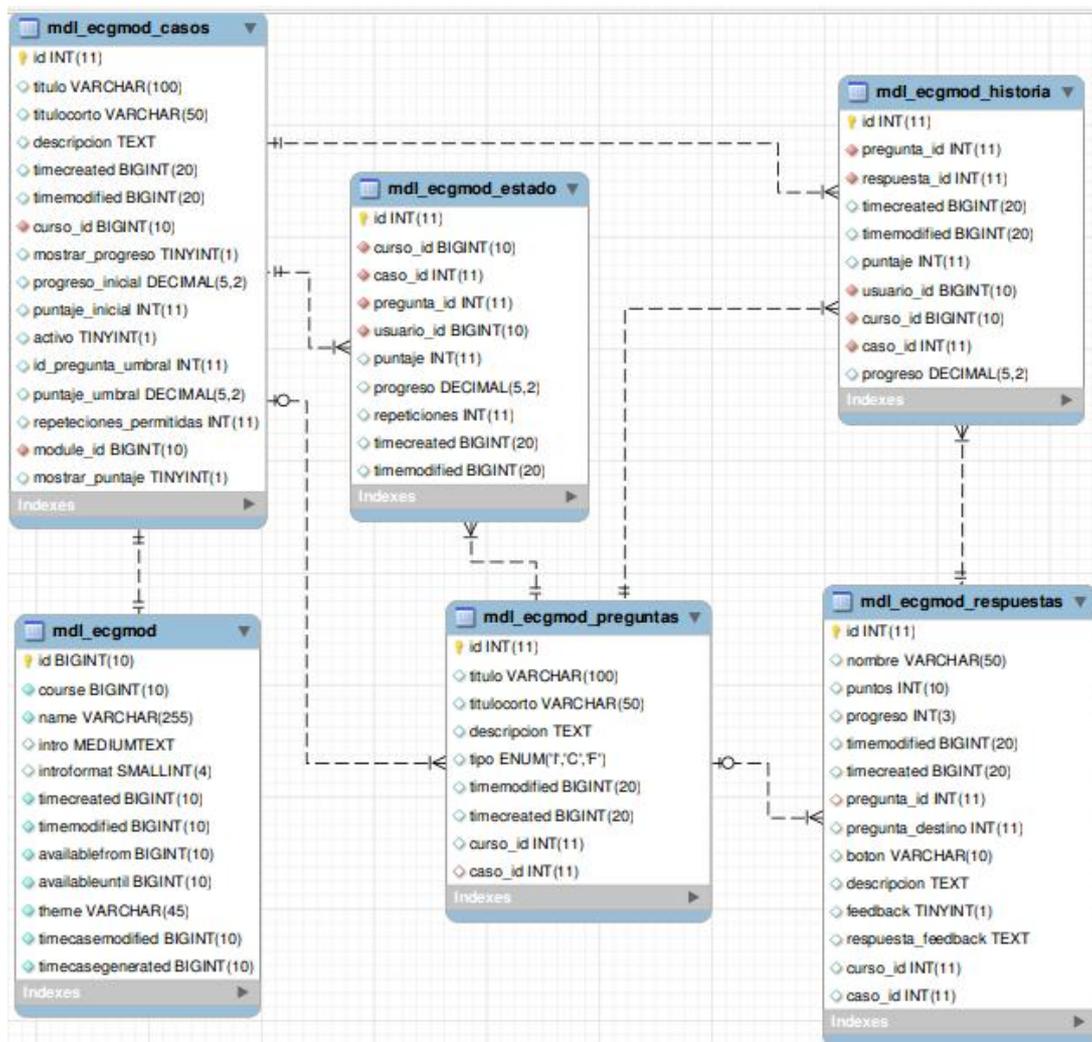


Figura 5.4 - MER - Modelo Entidad Relación

## 5.6. Dificultades encontradas

A continuación se detallan algunas de las dificultades que se fueron presentando en el transcurso del proyecto.

### 5.6.1. Escasez de documentación

Para los tres integrantes del grupo fue el primer enfrentamiento a Moodle como plataforma para desarrollo, por lo tanto, desde un comienzo se debió invertir tiempo de investigación sobre el funcionamiento interno de la plataforma, estilo de código, etc.

Se utilizó la documentación existente en el sitio de 'Documentación de Moodle',

particularmente la dirigida a desarrolladores, como por ejemplo el ‘Manual de estilo de la interfáz’. Sin embargo, la documentación allí encontrada resultó escueta teniendo en cuenta que recién se estaba ingresando al “mundo de desarrollo para Moodle”.

### **5.6.2. Compatibilidad entre navegadores**

Se utilizó HTML estándar, YUI para los CSS y jQuery para el desarrollo en JavaScript. A pesar que dichas tecnologías facilitan el desarrollo de código cross-browser, lograr una solución compatible entre los diferentes navegadores es una tarea que requiere un esfuerzo no despreciable. Se verificó que funcionara en los navegadores más comunes: Mozilla Firefox 3.5 o superior, Internet Explorer 8 o superior y Google Chrome.

### **5.6.3. Conocimiento en área de *e-learning***

Ninguno de los integrantes del grupo contaba con conocimientos ni experiencia en el área de *e-learning*, por lo que fue necesario adquirir los conocimientos necesarios en dicha área para lograr una solución adecuada al problema planteado.

### **5.6.4. Tecnologías utilizadas**

Dado que los integrantes del grupo no tenían conocimiento de algunas de las herramientas utilizadas tales como Moodle, PHP y JavaScript, fue necesario dedicar un tiempo considerable al estudio de las mismas.

## **5.7. Casos de estudio**

Se presenta en esta sección los casos de estudio que permiten validar la solución desarrollada en dos ejemplos concretos del problema a resolver: se implementaron dos interfaces de usuario que utilizan la API para presentar los ejercicios interactivos.

### **Interfaz 1 - Tabs**

Una de las interfaces implementadas utiliza la metáfora de *tabs* o *pestañas* para mostrar el caso al estudiante. A continuación mostramos una imagen de la interfaz:

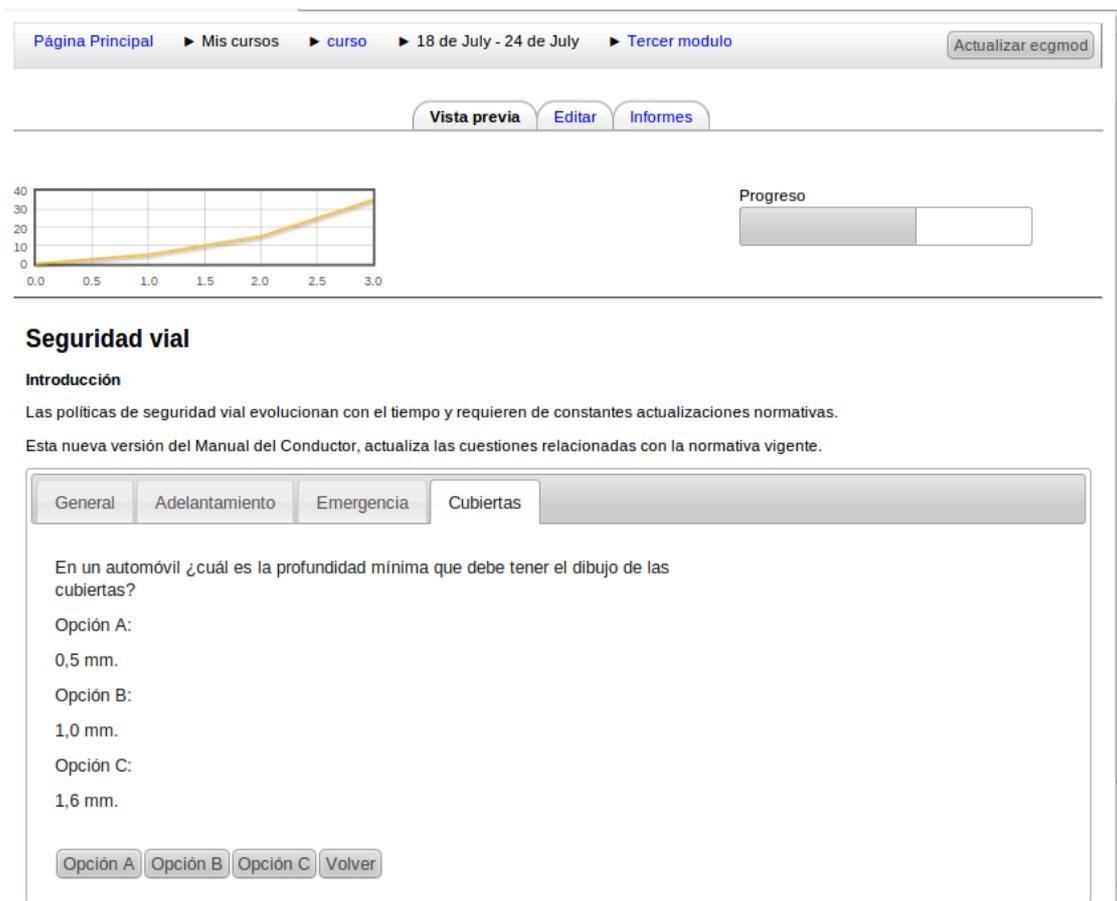


Figura 5.5 - Interfaz de Tabs

En la interfaz de *tabs*, cada escenario se presenta en un tab o pestaña. Se puede destacar que todo el contenido que muestra la interfaz es definido por el docente, en el módulo de Configuración.

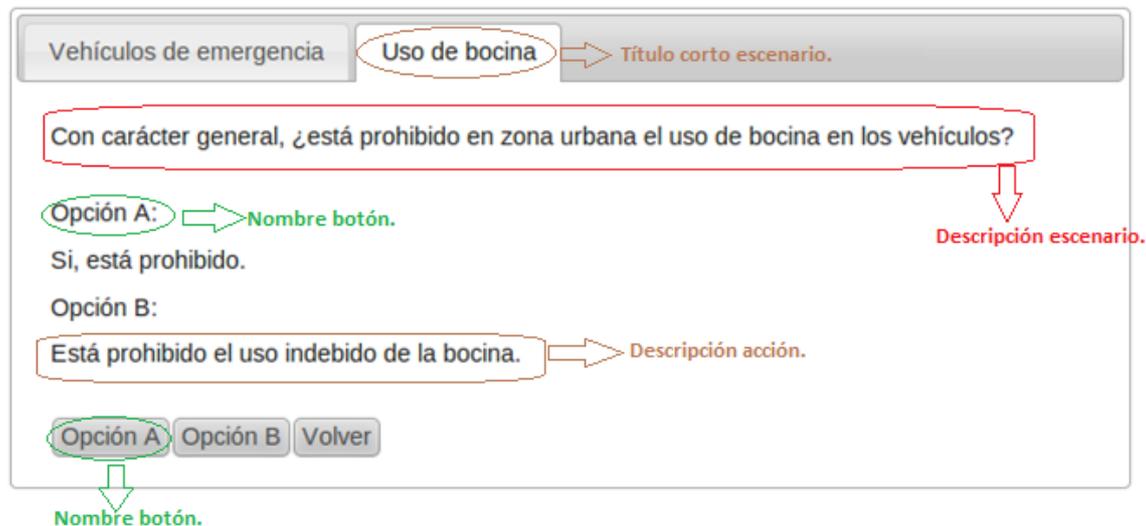


Figura 5.6 - Escenario

El nombre de la pestaña o tab es el título corto del escenario que define el administrador en la configuración del caso. Dentro del recuadro, en la parte superior, se encuentra la descripción del escenario, la cual puede contener elementos multimedia como vídeos, imágenes, etc. A continuación se muestra la descripción de cada acción, precedida por el nombre del botón como sub-título, es de notar que la descripción de la acción también puede contener elementos multimedia. Las mismas se van mostrando una a continuación de otra en forma vertical, con el nombre del botón como subtítulo.

Por último, en el recuadro de la parte inferior se ven los botones de las acciones para que el usuario estudiante seleccione una de ellas y pueda navegar el caso. Se muestra un botón por cada acción que tenga el escenario. A partir del segundo escenario, se genera el botón "volver", que permite al estudiante volver un escenario y seleccionar otra acción.

En la parte superior se presenta una gráfica de puntaje y una barra de progreso del caso. El usuario administrador configura el puntaje y el progreso que puede obtener el estudiante en cada acción que presenta el escenario.

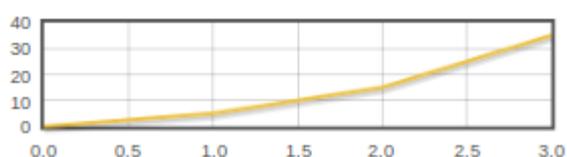


Figura 5.7 - Gráfica puntaje y progreso

El administrador puede configurar la posibilidad de mostrar al estudiante la gráfica de puntaje como también la barra de progreso. Ocultar estos elementos permite ganar espacio de pantalla para mostrar el resto del caso, característica que resulta útil en monitores de paqueño tamaño, como por ejemplo la XO.

En la parte inferior de la pantalla aparece en forma de vinculo la opción para que el estudiante pueda abandonar el caso. Si un estudiante abandona el caso, podrá retormarlo donde lo dejó la próxima vez que lo ejecute..

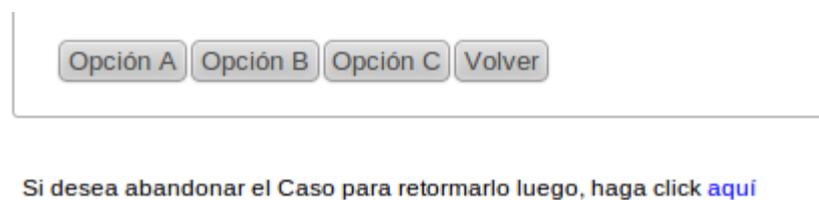


Figura 5.8 - Abandono del caso

La interfaz cuenta con la posibilidad de habilitar la visualización del feedback. Esto es realizado en el Módulo de Configuración por parte del docente, quien tiene la posibilidad de configurar a nivel de cada acción si se muestra feedback o no y el contenido de dica retroalimentación. Luego, cuando el estudiante selecciona una acción, se muestra el feedback en forma de ventana emergente. Esto permite brindar más información al estudiante, aclarar conceptos sobre la acción tomada o reforzar datos.

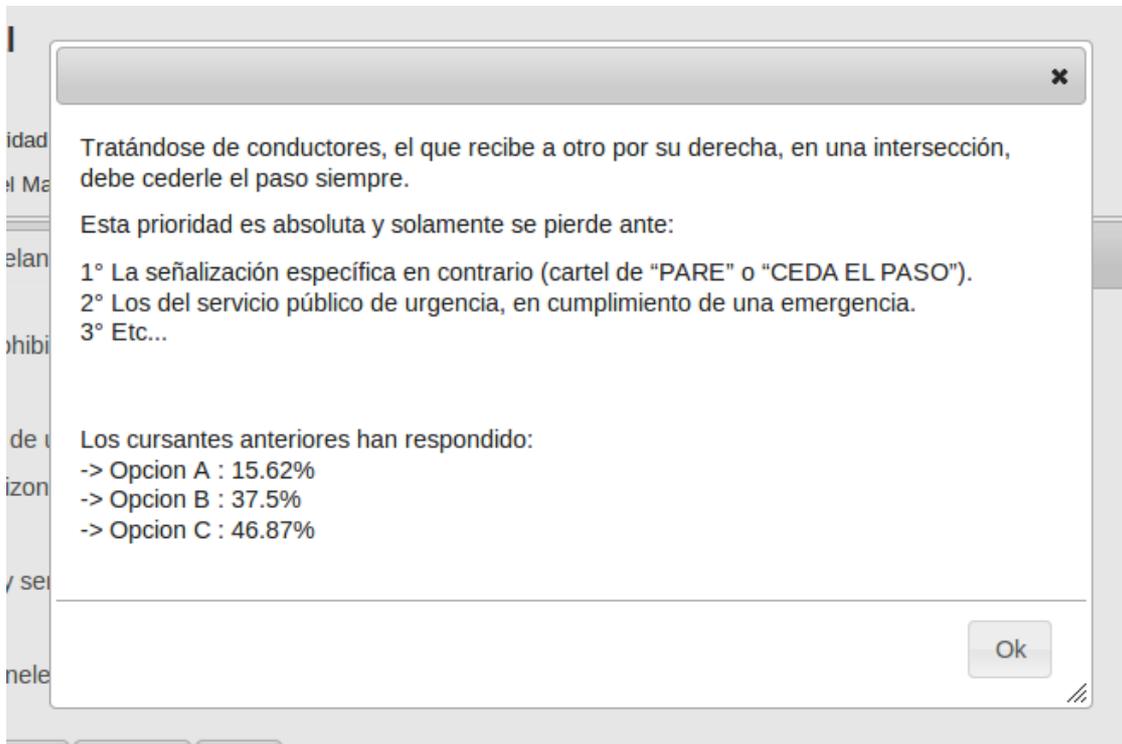


Figura 5.9 - Feedback

Como se ve en la figura anterior, en la parte inferior de la ventana de feedback, se muestran los porcentajes de selección de las diferentes acciones por parte de los estudiantes que ya han recorrido el caso.

## Seguridad vial

### Introducción

Las políticas de seguridad vial evolucionan con el tiempo y requieren de constantes actualizaciones normativas.

Esta nueva versión del Manual del Conductor, actualiza las cuestiones relacionadas con la normativa vigente.

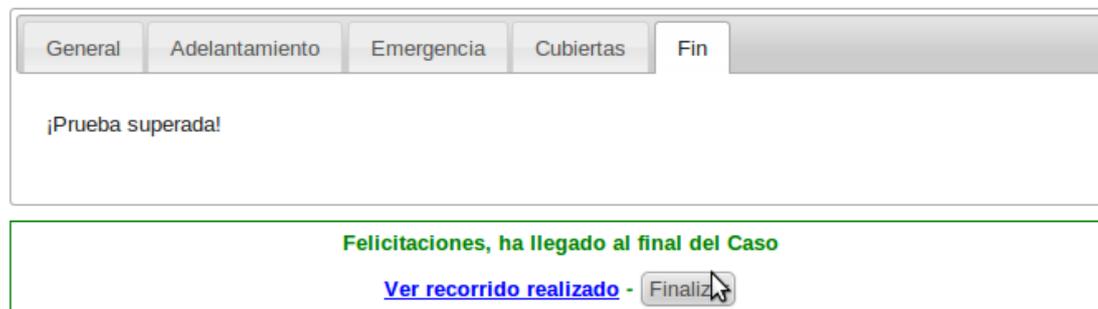


Figura 5.10 - Final del caso

Una vez que el estudiante llega al final del caso tiene la opción que le permite visualizar en un gráfico el recorrido que realizó. Esta opción aparece en forma de link con el texto "Ver recorrido realizado".

A continuación se muestra en forma de ventana emergente el recorrido realizado por el estudiante. Se puede apreciar todo el grafo del caso, en color rojo las acciones tomadas y los escenarios presentados.

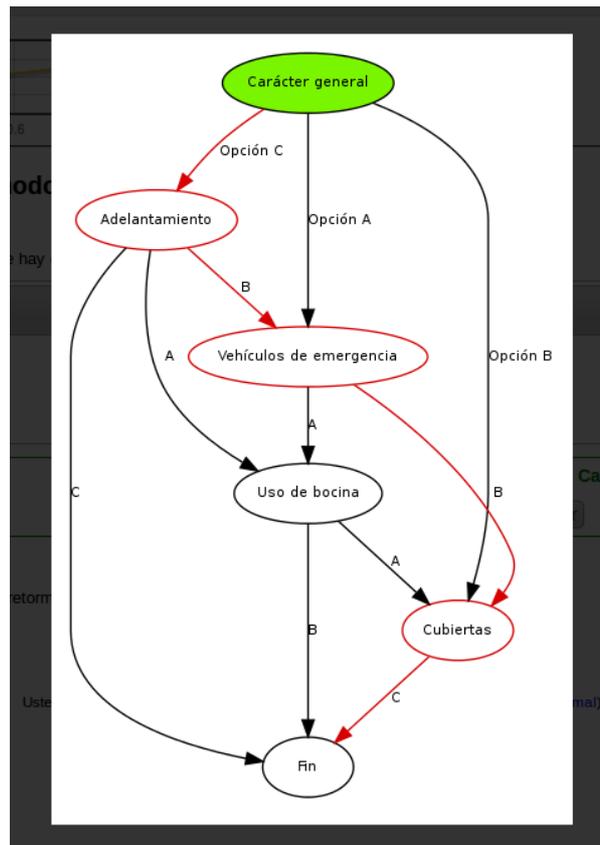


Figura 5.11 - Recorrido realizado

Una vez que finaliza el caso el sistema vuelve a la vista principal del curso.

## Interfaz 2 - Acordeón

La interfaz implementada utiliza la metáfora de *acordeón* o *desplegable* para mostrar el caso al estudiante. A continuación mostramos una imagen de la interfaz:



## Seguridad vial

### Introducción

Las políticas de seguridad vial evolucionan con el tiempo y requieren de constantes actualizaciones normativas.

Esta nueva versión del Manual del Conductor, actualiza las cuestiones relacionadas con la normativa vigente.

▶ **General**

▶ **Adelantamiento**

▼ **Emergencia**

Los vehículos de emergencia ¿Cuándo tienen prioridad de paso?

Opción A:

- Cuando están en una situación de emergencia.

Opción B:

- Cuando están en situación de emergencia y circulan por una avenida.

Si desea abandonar el Caso para retormarlo luego, haga click [aquí](#)

*Figura 5.12 - Presentación acordeón*

Tiene un comportamiento similar al de la interfaz de pestañas, con la diferencia que los escenarios se van presentando como un acordeón de paneles, donde al desplegar uno, se ocultan los demás.

La información es mostrada con el mismo criterio que en la interfaz de tabs. Por ejemplo el nombre de cada panel del acordeón es el título corto del escenario. Luego se presenta la descripción del escenario con las acciones y por último un botón para cada acción.

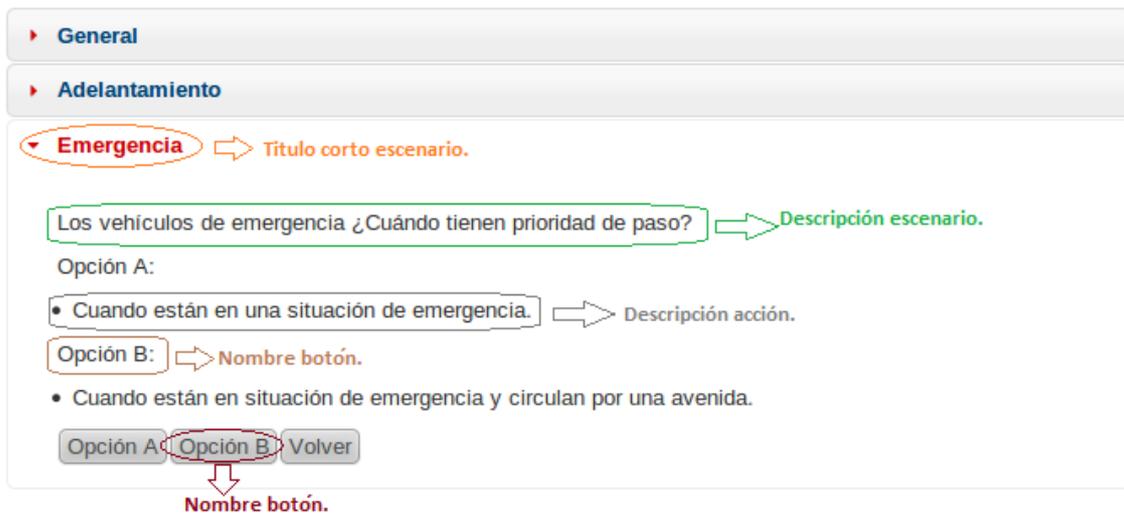


Figura 5.13 - Escenario acordeón

A partir del segundo panel del acordeón se muestra el botón volver y realiza el mismo comportamiento que en la interfaz anterior.

En el encabezado de página se maneja una gráfica de puntaje y una barra de progreso que se actualizan dependiendo de la acción tomada por el estudiante y de acuerdo a la configuración del caso.



Figura 5.14 - Cabezal interfaz

En el pie de pagina podemos encontrar la opción de abandonar el caso y luego cuando lo retome vuelve al mismo escenario que estaba el estudiante.

En las distintas acciones se maneja el feedback donde también se permite configurar si mostrarlo o no en el Módulo de Configuración del caso.

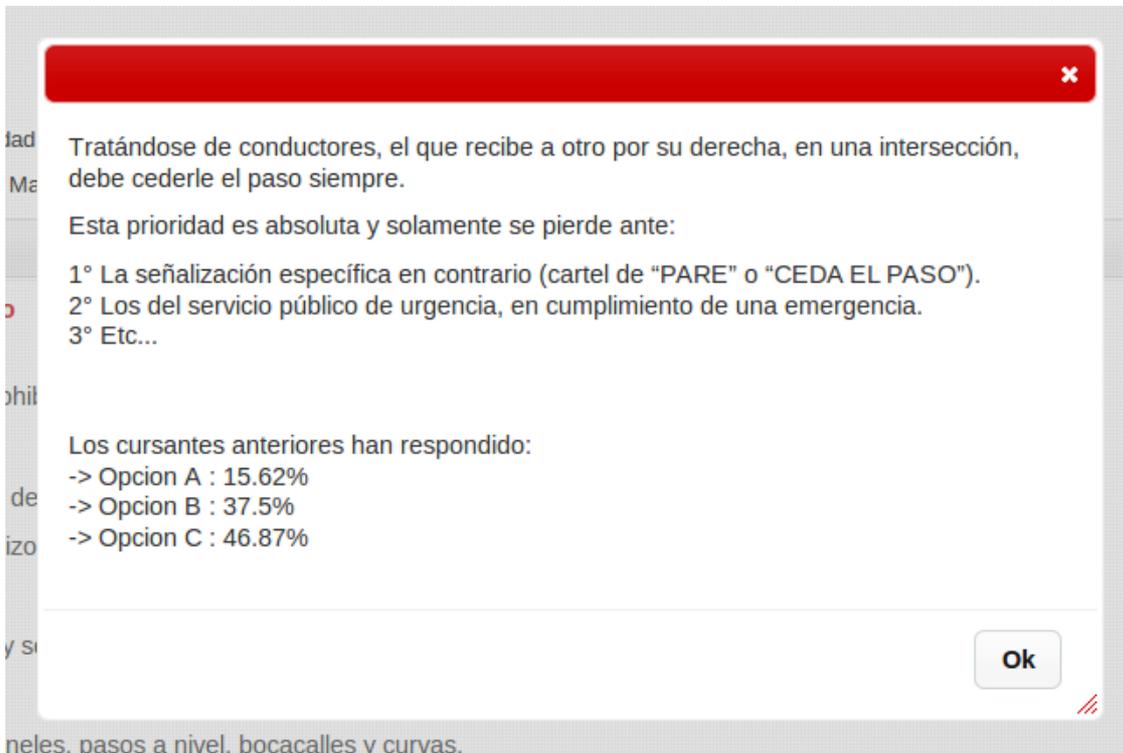


Figura 5.15 - Feedback acordeón

Las demás funcionalidades son idénticas a las que maneja la interfaz de *tabs*. Como por ejemplo, al finalizar el caso poder visualizar el recorrido que realizó el estudiante.

## 5.8. Testing

Se cumplieron con dos grandes etapas de un proceso de testing (Planificación y Ciclo de prueba) de forma paralela a varias de las etapas del proyecto. El objetivo planteado fue descubrir tempranamente los defectos para corregirlos y asegurarse de lograr un producto de calidad.

Como parte de la *planificación* se determinaron los tipos de pruebas que se realizarían (estáticas y/o dinámicas) y las herramientas necesarias para ello.

Respecto a la etapa de *ciclo de prueba*, se realizaron diferentes tipos de testing y en momentos distintos dependiendo de la funcionalidad que se quería probar. Se

ejecutaron *pruebas unitarias* durante todo el desarrollo y en algunos casos se realizaron *pruebas automatizadas* (scripts) y *pruebas no automatizadas* (revisión entre pares).

- Pruebas unitarias: debido a la agilidad de los lenguajes de programación utilizados (lenguajes dinámicos no compilados), muchas funcionalidades eran desarrolladas y testeadas inmediatamente, realizando un testing *manual*.
- Testing automatizado: se realizó en el último período de implementación, momento en el que se diseñaron, configuraron y ejecutaron las pruebas. La estrategia utilizada consistió en aplicar la técnica de caja negra, básicamente para realizar pruebas de regresión. La herramienta utilizada con este fin fue Selenium IDE (una extensión para Firefox), la cual permitió encontrar fallas y detectar las faltas que las provocaban. Para gestionar adecuadamente los defectos encontrados, se utilizó la herramienta Mantis que permite registrar y realizar el seguimiento de los incidentes.
- Testing no automatizado: se realizaron revisiones entre pares en pruebas de integración, verificación estática-manual, entre otros.

El detalle de las pruebas realizadas se puede ver en el *Anexo - Especificación del testing realizado*.

# Capítulo 6 - Gestión de proyecto

En esta sección se describe el proyecto desde el punto de vista de la gestión del mismo. Se hace referencia a la descripción del equipo y las tareas desempeñadas, la distribución de las actividades del proyecto y las herramientas de gestión utilizadas.

## 6.1. Descripción del equipo y tareas desempeñadas.

El equipo está compuesto por tres integrantes, los que no contaban con experiencia en todas las herramientas utilizadas. A medida que transcurrió el tiempo se fueron adquiriendo conocimientos en las diferentes herramientas y aplicando ese conocimiento en beneficio del proyecto. Los roles de los integrantes fueron variando a medida que se avanzó en el mismo y los integrantes trabajaron en todas las áreas para distribuir de forma homogénea el trabajo.

Comentarios:

- Al comienzo del proyecto, el análisis del problema y el estudio de las diferentes herramientas y tecnologías fue realizado por todos los miembros del grupo, con ayuda del tutor.
- Luego de definida la arquitectura a alto nivel, se dividieron los roles en función de los siguientes criterios generales:
  - desarrollo del configurador de casos (integración con Moodle)
  - desarrollo del generador de casos
  - desarrollo de las interfaces
  - integración interfaz / generador

Dependiendo de la etapa del proyecto y del esfuerzo necesario para avanzar en tiempo y forma con cada uno de los componentes del sistema, cada integrante fue rotando por los distintos roles, lo cual permitió que todo el grupo contara con conocimiento suficiente de cada uno de los componentes de desarrollo.

A continuación se presenta un resumen de las horas invertidas en las diferentes áreas del proyecto:

**IMPORTANTE: ESTA SUMADO SOLO HASTA SETIEMBRE**

Área	Horas Totales
Investigación / Capacitación	69
Prototipo	105
Instalación de herramientas / ambiente	67.5
Desarrollo	240.5
Coordinación	57.5
Reuniones con tutor	62.5
Documentación	132
Análisis	85
Prototipado	114
Desarrollo y Testing	83

*Tabla – Distribución horaria*

## 6.2. Herramientas de gestión

Para el correcto seguimiento del proyecto, gestión de las tareas y control del avance, se usaron algunas herramientas que facilitaron dichas tareas. Al comienzo del proyecto se probaron varias alternativas, para luego definir las que resultaron más prácticas.

### **Google Sites:**

Se comenzó publicando un sitio web utilizando Google Sites, y algunas de sus

herramientas para el ingreso de tareas y publicación de documentos. Esta herramienta no resultó práctica, por lo que se pasó a utilizar otras alternativas. Las razones por las cuales no fue de gran practicidad fueron básicamente que no contaba con notificaciones cuando se realizaban modificaciones y/o actualizaciones y que la gestión de tareas no tiene todos los filtros necesarios para gestionarlas de forma eficiente.

URL del sitio: [33]

**Dropbox:**

Dropbox [30] es una herramienta que permite sincronizar archivos en diferentes computadoras. Además permite compartir carpetas entre diferentes personas y mantener las carpetas sincronizadas. Se utilizó para compartir documentos entre los integrantes del grupo. Dropbox mantiene además un versionado de los documentos, permitiendo restaurar versiones pasadas y saber qué persona realizó cada cambio. Dropbox resultó de una enorme utilidad a la hora de gestionar archivos.

**Balsamiq:**

Balsamiq [31] es una herramienta que permite realizar mockups online de forma sencilla y colaborativa. Se utilizó para boctar las diferentes interfaces.

**Mantis:**

Mantis [32] es una herramienta de bug tracking gratuita. Fue utilizada para la gestión de tareas y el seguimiento de bugs. Mantis permite registrar “incidentes”, adjuntar archivos, agregar notas, asignarlos a personas y realizar el seguimiento del estado de cada incidente. De esta forma, el equipo tenía una pila de tareas, las cuales se asignaban en función de los roles y se podía tener un panorama claro de la cantidad de incidencias pendientes en cada momento. Además, la herramienta permite clasificar las incidencias en diferentes categorías, como por ejemplo, “error”, “mejora”, “error crítico”, etc y asignar prioridades.



Logged in as: nicolasm (Nicolas Medeiros - developer)

2011-11-19 14:09 ART

<a href="#">Main</a>   <a href="#">My View</a>   <a href="#">View Issues</a>   <a href="#">Report Issue</a>   <a href="#">Change Log</a>   <a href="#">Roadmap</a>   <a href="#">My Account</a>   <a href="#">Logout</a>	
<b>Assigned to Me (Unresolved) [ ^ ] ( 1 - 3 / 3 )</b> 0000037  documentacion: desarrollo de themes [All Projects] General - 2011-10-14 20:03 0000031  grafo: nodo con multiples acciones [All Projects] Mejora - 2011-10-08 08:41 0000009  Estudiar como funcionan los GRADES y GRADEBOOK en Moodle [All Projects] Mejora pedida tutor - 2011-09-03 17:41	<b>Unassigned [ ^ ] ( 1 - 2 / 2 )</b> 0000038  consulta: se podria pasar el lenguaje actual a la interfaz? [All Projects] Mejora - 2011-10-16 10:17 0000017  documentacion [All Projects] General - 2011-09-03 16:17
<b>Reported by Me [ ^ ] ( 1 - 10 / 22 )</b> 0000038  consulta: se podria pasar el lenguaje actual a la interfaz? [All Projects] Mejora - 2011-10-16 10:17 0000018  probando el script de instalacion [All Projects] General - 2011-10-16 10:05 0000037  documentacion: desarrollo de themes [All Projects] General - 2011-10-14 20:03	<b>Resolved [ ^ ] ( 1 - 10 / 74 )</b> 0000018  probando el script de instalacion [All Projects] General - 2011-10-16 10:05 0000033  chequeo de validez del caso por fecha cuando desactivo la disponibilidad [All Projects] Bug - 2011-10-10 16:15 0000029  Cuando usuario ingresa y tiene estado guardado, consultarle si quiere [All Projects] Funcionalidad - 2011-10-09 18:00

Figura 6.1 - Captura de pantalla del Mantis de uno de los integrantes

El uso de Mantis fue de gran utilidad para el correcto seguimiento de las tareas.

## 6.3. Herramientas de desarrollo

Para desarrollar el proyecto se utilizaron las siguientes herramientas:

### PHP:

PHP es el lenguaje de programación interpretado, dinámico y débilmente tipado, que permite utilizar técnicas de programación orientada a objetos, con sintaxis derivada de C y que es ejecutado del lado del servidor. Dado que Moodle fue implementado utilizando PHP, se utilizó esta tecnología para el desarrollo del módulo. La versión utilizada es la 5.3, ya que es la versión requerida por Moodle 2, siendo además la versión más reciente de este lenguaje.

### JavaScript y jQuery:

JavaScript es un lenguaje de programación interpretado, orientado a objetos y dinámico, que se ejecuta del lado del cliente (aunque existen implementaciones de servidores que permiten ejecutar código JavaScript). Permite mejorar las funcionalidades de la interfaz web, mediante la manipulación del DOM (Document Object Model) y el manejo de eventos.

jQuery [17] es una biblioteca JavaScript, que permite simplificar la manera de manipular el DOM, manejar los eventos, implementar técnicas AJAX, entre otras cosas. Por ejemplo, para seleccionar un elemento del DOM utilizando su *id*:

Utilizando JavaScript:

```
document.getElementById(identificador)
```

Utilizando jQuery:

```
$("#identificador")
```

El ejemplo es una muestra de las ventajas ofrecidas por jQuery para agilizar el desarrollo en JavaScript.

Otra de las ventajas de la utilización de jQuery es la enorme cantidad de plug-ins disponibles para todo tipo de situaciones: jQuery UI para crear componentes de interfaz de usuario (selectores de fecha, ventanas modales, indicadores de progreso, etc),

#### **Eclipse IDE:**

Como editor de código se utilizó Eclipse [21] con el plug-in PHP Development Tools [22] para ayuda de código PHP.

#### **MySQL y MySQL Workbench:**

El motor de base de datos utilizado fue MySQL [23]. Para su administración se utilizó MySQL Workbench [24]

#### **SVN:**

Para versionado de código se utilizó SVN, hosteado en XP-DEV [25]. Los clientes SVN utilizados fueron RapidSVN en Linux [26] y TortoiseSVN en Windows [27]

#### **Selenium IDE:**

Se utilizó Selenium IDE [28] para automatizar pruebas básicas sobre las distintas interfaces y contar con una verificación simple, comprobando así que el sistema funcione adecuadamente.

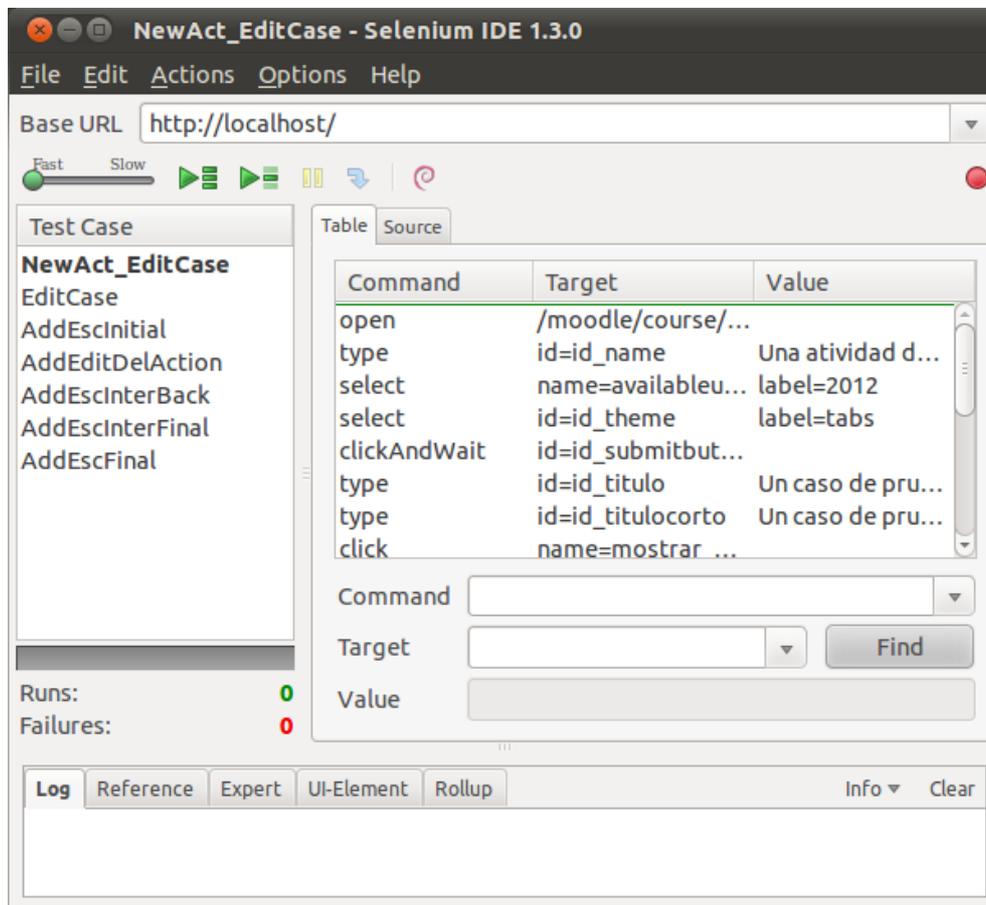


Figura 6.2 - Selenium IDE

Selenium IDE es una extensión de Firefox que brinda un entorno integrado para la administración de scripts de testeo, la cual permite grabar, editar, depurar y ejecutar los tests.

#### **Firebug:**

Firebug [29] es una extensión de Firefox que resulta esencial para la corrección de problemas con estilos CSS, código HTML y aplicaciones AJAX.

## **6.4. Desarrollo del proyecto**

El proyecto surge como iniciativa de la empresa EviMed, la cual brinda “consultoría especializada en servicios de educación médica continua y desarrollo de software para el área médica”. Con la propuesta realizada se pretendía lograr una nueva herramienta que permitiera a los docentes que trabajan con la plataforma Moodle generar

contenidos educativos interactivos.

El principal objetivo planteado al comienzo del proyecto fue el desarrollo de un módulo para la plataforma Moodle, que permitiera generar casos interactivos arbitrarios y que dichos casos pudiesen ser presentados de diferentes maneras. Además era necesario que la interacción entre la interfaz y el servidor fuera mínima, de forma tal que la experiencia del usuario resultara lo más fluida posible.

Desde el comienzo se tuvo como prioridad finalizar el proyecto dentro del período de nueve meses, por lo que se definió un cronograma con dicho objetivo y se trabajó en forma continua para cumplir con los plazos fijados. Para esto, el equipo se reunió varias veces por semana durante todas las semanas que abarcó el mismo. De esta forma, se mantenía un control estricto sobre el avance del proyecto, el cumplimiento de las tareas asignadas y se coordinaban los pasos a seguir para la siguiente semana. Además, se realizaron reuniones quincenales con el tutor para mostrar el avance, recibir feedback y definir las correcciones y mejoras a realizar.

Se comenzó el proyecto investigando diferentes propuestas de educación a distancia y sus características. Se realizaron reuniones con el equipo de EviMed para conocer más acerca de temas como Educación Médica Continua, educación a distancia y el uso de tecnologías de la información y comunicación para la educación. Además se comenzó a estudiar la plataforma Moodle, su funcionamiento, filosofía y el código del sistema.

Durante los primeros dos meses se investigó el estado actual de las plataformas de educación a distancia, las herramientas brindadas por Moodle y cómo podrían mejorarse. Se desarrollaron diferentes prototipos para probar interfaces alternativas y su correcto funcionamiento en diferentes clientes: navegadores, funcionamiento en las XO y en los sistemas operativos Linux y Windows.

Luego de haber instalado el ambiente de desarrollo se comenzó a trabajar en los diferentes componentes del módulo: configurador de casos, generador y API, y las interfaces web. El desarrollo del módulo de configuración insumió varias horas de estudio del código de Moodle y de los estándares de su codificación. Dado que se espera liberar el código del módulo a la comunidad Moodle, se desarrolló tomando en

cuenta el estándar definido para el desarrollo de módulos.

A medida que se fue avanzando en las etapas del proyecto, se tomaron diferentes decisiones sobre lo que se debería producir, algunas de las decisiones más relevantes se detallan a continuación:

- Utilizar la versión 2.0.3 de Moodle en favor de la 1.9. Esta decisión se tomó luego de estudiar el código y funcionalidades de ambas versiones. Ver comparativa sección 2.3.5.
- Alta integración con Moodle: el modulo haría uso de las herramientas brindadas por la plataforma: acceso a datos, generación de HTML, multilinguaje.
- Utilizar JSON como formato para intercambio de datos entre la API y el cliente.
- Interfaces a desarrollar: se decidió que únicamente se desarrollarían interfaces utilizando HTML/Javascript/CSS. Se realizó un prototipo en Flash para validar el uso de la API y la validez de JSON como tipo de datos para ser consumido desde la aplicación Flash.

Un conjunto más amplio y detallado de las decisiones tomadas a lo largo del proyecto se detalla en la sección '4.5. Decisiones tomadas'.

Cuando surgió la posibilidad de presentarse a la conferencia MoodleMoot Uruguay 2011 se tomó la decisión de enfocarse en la preparación de la presentación para dicha conferencia. Por lo tanto, se fijó una fecha temprana para tener el mayor avance posible en cuanto al desarrollo, lo que dio un impulso fundamental para afrontar las últimas etapas del proyecto. Dicha preparación y presentación para la conferencia brindó una instancia temprana de autoevaluación del estado del proyecto.

El mecanismo de trabajo con el tutor consistió en realizar reuniones de seguimiento cada quince días, a las que asistían los 3 integrantes del grupo, el tutor y otros miembros de la empresa cliente. Esto permitió hacer un seguimiento constante del avance, rectificando las acciones y decisiones tomadas de forma temprana.

## **6.5. Desarrollo de actividades en el proyecto**

Al comienzo del proyecto se definió un cronograma global tentativo a seguir a lo largo del año. Junto a este cronograma, se elaboró un diagrama de Gantt para planificar la secuencia de actividades que compondrían el proyecto. Dicho diagrama se fue modificando a medida que se avanzaba en el mismo, ya que surgieron nuevos elementos como por ejemplo la conferencia MoodleMoot, que obligaron a realizar cambios en la agenda. De todas formas, el diagrama resultó de mucha utilidad para tener una visión global del avance según la planificación.





*Figura 6.4 - Diagrama de Gantt segunda parte*

El Plan de trabajo planteado inicialmente era el siguiente:

Mes 1: investigación de herramientas existentes, acercamiento a usuarios, médicos y organizaciones para conocer sus necesidades, estudio de tecnologías que permitan lograr los objetivos (matriz con análisis de ventajas y desventajas de cada herramienta). Con esto, crear un estado del arte en la materia.

Mes 2: propuesta de la arquitectura de la solución, y documentación que la explique y justifique. Tomar la decisión de qué tecnologías utilizar (asegurando la misma a través de un prototipo técnico) y una idea macro de qué funcionalidades se desarrollarán, a partir de nuevas reuniones de relevamiento. Creación del plan de desarrollo. Ajustes a la arquitectura; diseño de la base de datos. Especificación de servicios a brindar por la herramienta. Configuración de entorno de desarrollo y adaptación a la misma y a las tecnologías. Documentación de las decisiones tomadas.

Mes 3: ajustes al plan de desarrollo teniendo en cuenta las tecnologías seleccionadas, al ambiente de desarrollo, arquitectura del sistema y funcionalidades identificadas. Documentación de las decisiones tomadas. Cerrar el alcance del producto con todos los involucrados. Comienzo de la implementación.

Meses 4, 5, 6 y 7: desarrollo iterativo incremental de la solución, apoyados en un fuerte ida y vuelta con los diferentes tipos de usuarios: relevamiento y pruebas. Ajustes sucesivos a la planificación, funcionalidades y servicios definidos. Documentación de lo realizado, los resultados obtenidos y las decisiones tomadas.

Meses 8 y 9: Pruebas unitarias, integrales y de stress. Dos pruebas piloto con participación del cliente. Ajustes finales al desarrollo. Utilización en un ambiente de Producción. Publicación en evento científico.

Mes 10: consolidación de la documentación final (definiendo en esta etapa los trabajos a futuro) y validación de lo realizado. Preparación de la presentación final y defensa.

Como puede verse, se logró ajustarse al plan de trabajo, minimizando desvíos, de forma de cumplir con el cronograma planteado

## 6.6. Logros obtenidos

Se considera que los resultados obtenidos en el proyecto cumplen con los objetivos planteados al comienzo del mismo, logrando desarrollar un nuevo módulo que aporte nuevas funcionalidades y opciones a la plataforma Moodle. Se logró definir un mecanismo para independizar la presentación al estudiante de los ejercicios de Moodle, ofreciendo una API para el consumo y manipulación de dichos ejercicios.

Como evaluación del módulo desarrollado, se puede destacar que:

- permite instalarse, actualizarse y desinstalarse de la misma manera que cualquier otro módulo de Moodle.
- se cumplió en varios aspectos con el estándar de codificación especificado en el *manual de estilo de código* de Moodle [\*].
- se utilizaron las herramientas brindadas por Moodle para el acceso a datos, soporte multi-idioma, elementos para la interfaz de usuario, manejo de permisos y ayuda en línea.
- es posible combinar el módulo desarrollado con otras funcionalidades de Moodle, tales como permisos de grupos y períodos de vigencia.
- cumplimiento del plan de trabajo planteado originalmente, y del cronograma establecido.

Otro objetivo que se logró cumplir fue presentar los resultados del proyecto en un congreso, en particular en el congreso MoodleMoot Uruguay 2011, ante un público de diferentes países quienes mostraron interés en las posibilidades que ofrece el módulo.

En resumen, gracias al mecanismo de reuniones quincenales que permitieron un avance constante, rectificando las acciones y decisiones de forma temprana, y los logros mencionados anteriormente, se considera que se lograron cumplir los objetivos

y expectativas planteadas inicialmente por el tutor e integrantes del grupo.

## 6.7. Aportes del proyecto

Los siguientes ítems consideran los principales aportes a destacar dentro de la evaluación del proyecto:

- Relevamiento de herramientas disponibles para la generación de ejercicios interactivos en la plataforma Moodle.
- Comparación con otros ejercicios interactivos utilizados por diferentes organizaciones educativas, con foco en ofrecer los mecanismos adecuados para el correcto aprovechamiento de los cursos por parte de los estudiantes y docentes.
- Un nuevo módulo que será liberado a la comunidad Moodle. Se construyó un módulo que quedará disponible para ser utilizado por los educadores de forma gratuita y que es una base para continuar agregando funcionalidades que la propia comunidad vaya identificando como necesarias.
- Una herramienta que permite independizarse de la interfaz de Moodle para implementar diferentes interfaces que se adapten mejor a cada caso en particular. Esto además permite utilizar un mismo módulo para diferentes aplicaciones, como juegos educativos, ejercicios, cuestionarios, etc.
- Haber participado del encuentro MoodleMoot Uruguay 2011[12] realizado en la Facultad de Ingeniería, presentando el nuevo módulo y logrando una buena recepción de los participantes.
- Validación de la integración de varias tecnologías con la plataforma Moodle, principalmente desde el punto de vista de la interfaz del alumno en donde se utilizó HTML, JavaScript, Ajax; además se logró integrar una herramienta (Graphviz) para visualizar el grafo correspondiente al Caso en cuestión.

## 6.8. Aprendizajes

En esta sección se comentan las enseñanzas más importantes que ha dejado el proyecto, junto con comentarios sobre sus aspectos más importantes.

A lo largo del proyecto se fueron incorporando experiencias y enseñanzas, algunas de las cuales merecen ser destacadas:

- Se estudiaron y se pusieron en práctica los conceptos de programación web asincrónica (AJAX). Para implementar interfaces dinámicas utilizando AJAX se utilizaron las bibliotecas jQuery y YUI (Yahoo User Interface). Se trabajó con conceptos como el DOM (Document Object Model) en páginas web y se utilizaron diferentes plug-ins de jQuery para la implementación de las interfaces gráficas. Algunos de los plug-ins utilizados fueron:
  - jFlot: para realizar gráficas dinámicas utilizando JavaScript.
  - jQuery UI: se utilizó la biblioteca para interfaces gráficas de jQuery para implementar las ventanas modales, la barra de progreso, la interfaz de pestañas y el acordeón.
  - YUI versión 2 para las hojas de estilo CSS. Esto permite “resetear” los estilos de los navegadores a un estilo común y conocido, además de crear grillas CSS de forma sencilla
- Experiencia obtenida como resultado de aplicar conocimientos adquiridos en el transcurso de la carrera en las distintas etapas del proyecto, tanto desde el punto de vista del desarrollo como de la gestión del proyecto. Los conocimientos en áreas tales como Planificación, Alcance (Requerimientos), Diseño de la solución, entre otros, fueron de vital importancia para cumplir con los objetivos planteados.
- Conocimientos adquiridos a partir del estudio y la utilización de distintas herramientas durante el proyecto, tanto a nivel de desarrollo (lenguajes de programación, gestión de la configuración, etc.) como de gestión del proyecto. Para más información sobre las herramientas utilizadas, ver sección *Gestión del proyecto*.

- Acercamiento a la plataforma Moodle y su comunidad, a través de la investigación sobre el funcionamiento interno de la herramienta y la experiencia recogida como participantes del evento MoodleMoot Uruguay 2011, el cual reunió a investigadores, docentes, estudiantes, desarrolladores y administradores de dicha plataforma de educación.

La presentación realizada está disponible en el sitio de la herramienta utilizada Prezi [5] y se pueden ver los videos de demo utilizados en You Tube ([6] y [7]).

## Capítulo 7 - Trabajos a futuro

Los puntos aquí planteados pueden considerarse como una guía para mejoras futuras de la solución obtenida en este proyecto. Al ser la solución obtenida un módulo que cumple con las guías y estándares de Moodle y además permite implementar diferentes interfaces para los ejercicios, deja abierta la posibilidad de agregar nuevas funcionalidades y encontrar nuevos usos para la herramienta.

Se identificaron los siguientes trabajos a futuro:

### 1. Integración a la instalación base de Moodle.

Trabajo a realizar:	Presentación del módulo como posible extensión de la plataforma, con el objetivo de ser aceptado como “módulo oficial” por la comunidad Moodle.
Situación actual:	Un módulo independiente que ha sido desarrollado siguiendo los estándares para la integración.

### 2. Personalización de reportes estadísticos.

Trabajo a realizar:	Realización de reportes específicos sobre estadísticas de los cursos.
Situación actual:	Se realizó un reporte general.

### 3. Presentación en Flash.

Trabajo a realizar:	Diseñar e implementar una interfaz de presentación del caso utilizando Flash, consumiendo las operaciones que brinda la ‘Biblioteca de consumo’.
Situación actual:	Se realizó un prototipo utilizando Flash para validar su integración.

### 4. Reutilización de *Escenarios* entre los distintos *Cursos*.

Trabajo a realizar:	Permitir en un <i>Caso</i> utilizar los <i>Escenarios</i> de los demás <i>Casos</i> definidos en el sistema.
Situación actual:	En un <i>Caso</i> solamente se puede utilizar los <i>Escenarios</i> definido en ese <i>Caso</i> .

### 5. Copiar Caso

Trabajo a realizar:	a Implementar la posibilidad de copiar un caso, con sus escenarios y acciones. Esto facilita la generación de nuevos casos.
Situación actual:	N/A

### 6. Algoritmo de grafo.

Trabajo a realizar:	Utilizar un algoritmo para calcular de forma automática el porcentaje de avance y el puntaje en los escenarios para todos los caminos.
Situación actual:	N/A

### 7. Integrar las Clasificaciones de Moodle.

Trabajo a realizar:	Permitir que el módulo interactúe con el mecanismo de clasificaciones utilizado por Moodle.
Situación actual:	N/A

### 8. Implementar más reportes.

Trabajo a realizar:	Agregar más reportes que sean de utilidad para los docentes.
Situación actual:	Se cuenta con un reporte genérico.

## Capítulo 8 - Conclusiones

Las siguientes son algunas conclusiones alcanzadas a lo largo del desarrollo del proyecto.

En primer lugar, se destaca haber cumplido con éxito el desarrollo de la herramienta para la generación de casos interactivos integrada a Moodle. Se siguieron los estándares de Moodle para el desarrollo del módulo, lo que permitirá liberarlo a la comunidad. Se logró un bajo acoplamiento entre la presentación de los casos y la configuración de los mismos, facilitándose la creación de diferentes interfaces y dando libertad de poder presentar los casos de múltiples formas.

Además, se realizó de forma exitosa la presentación de la herramienta en el encuentro internacional MoodleMoot (12) realizado en Octubre del 2011 en las instalaciones de Facultad de Ingeniería, con participantes de diferentes países como Brasil, Venezuela, Perú, entre otros. Alguno de los asistentes a la presentación mostraron interés en el trabajo, informándose sobre la posibilidad de utilizar el módulo cuando sea liberado. A continuación se muestra la imagen de la publicación del artículo.

#### Sesión 6. Prácticas Tecnológicas.

Paper 52: Integración del Repositorio de Objetos de Aprendizaje de AMBAR con la Plataforma Moodle  
Vanessa Miguel, Yosly Caridad Hernández Bieliukas, Irian Joubert, Elizabeth Ramírez y Nora Montaña.

paper

Paper 57: Composite Evaluation Process Interface Moodle

Gabriela Martins Gonçalves De Oliveira y Plinio Thomaz Aquino Junior.

paper

Paper 60: MoDaWeEd: un framework que integra Moodle, Data Mining y Web Usage Mining en el ámbito de la Educación

Esther Hochsztain y Andrómaca Tasistro.

paper

Paper 26: Generador de casos interactivos

Nicolás Medeiros, Guillermo Perez y Nicolás Sniadover.

paper

Paper 63: Error: Database connection failed.

Jose Rama.

paper

*Figura 8.1 - Presentación*

Con el desarrollo del proyecto se obtuvieron conocimientos de PHP, MySQL, Moodle, jQuery/CSS/HTML. Además, se adquirieron nuevos conocimientos en las tecnologías utilizadas para el diseño y desarrollo de la solución.

La empresa EviMed que colaboró activamente demostró su gran conformidad por haber alcanzado el desarrollo de la herramienta con éxito y haber cumplido con los objetivos solicitados.



## Capítulo 9 - Glosario

**RIA - Rich Internet Applications (Wikipedia - 13/11/2011):** Las rich Internet applications, o RIA (en español "aplicaciones de Internet enriquecidas"), son aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio. Las RIA surgen como una combinación de las ventajas que ofrecen las aplicaciones web y las aplicaciones tradicionales y buscan mejorar la experiencia del usuario. Normalmente en las aplicaciones web, hay una recarga continua de páginas cada vez que el usuario pulsa sobre un enlace. De esta forma se produce un tráfico muy alto entre el cliente y el servidor, llegando muchas veces a recargar la misma página con un cambio mínimo. En los entornos RIA, en cambio, no se producen recargas de página, ya que desde el principio se carga toda la aplicación, y sólo se produce comunicación con el servidor cuando se necesitan datos externos como datos de una base de datos o de otros ficheros externos.

**AJAX (Asynchronous JavaScript And XML, JavaScript asíncrono y XML):** es una técnica de desarrollo utilizada para crear aplicaciones RIA. Consiste en realizar llamados asincrónicos desde el cliente al servidor, sin requerir de una recarga completa de la página web. Esto permite actualizar sólo ciertas secciones de la página, o salvar información sin tener que recargar la página completa luego de realizar la acción.

**GNU (GNU is Not Unix):** Es un proyecto iniciado por Richard Stallman con el objetivo de desarrollar un sistema operativo totalmente compatible con Unix, pero libre y gratuito. Para esto se diseñó la licencia GPL que garantiza los derechos de ejecutar, copiar y modificar libremente el código de las aplicaciones.

**LDAP (Lightweight Directory Access Protocol):** es un protocolo de aplicación que permite acceder a un servicio de directorio para buscar y obtener información en una red.

**POP3 (Post Office Protocol):** Es un protocolo de aplicación utilizado por el cliente para obtener los mensajes de correo de un servidor. Sólo permite obtener los mensajes, pero no enviar.

**IMAP** (Internet Message Access Protocol) : es un protocolo de red para acceso a correos electrónicos almacenados en un servidor, que busca mejorar las prestaciones brindadas por POP3.

**NNTP** (Network News Transport Protocol): es un protocolo creado para leer y publicar noticias en Usenet (Users Network).

**SSL** (Secure Sockets Layer) y **TLS** (Transport Layer Security): son protocolos criptográficos utilizados para transmitir datos de forma segura a través de redes de computadores.

**Mock, mockups:** son *sketchs* o bocetos utilizados para diseñar primeras aproximaciones a diferentes soluciones.

**Background:** indica que un proceso se ejecuta en segundo plano, sin que el usuario note tal actividad.

# Capítulo 10 - Referencias

1. New England Journal of Medicine. <http://www.nejm.org/>  
Ejemplo de caso: <http://www.nejm.org/doi/full/10.1056/NEJMimc0806234>
2. MedScape <http://www.medscape.org>
3. BOU BAUZÁ, G. (2003). El guión multimedia. España: Anaya Multimedia. 2003. ISBN: 8441514593 ISBN-13: 9788441514591
5. Prezi, presentación MoodleMoot Uruguay 2011.  
<http://prezi.com/capjonwloglw/generador-de-casos-interactivos/>
6. Video demostración utilizado en presentación MoodleMoot Uruguay 2011.  
<http://www.youtube.com/watch?v=j4rRYmDHbzw>
7. Video demostración utilizado en presentación MoodleMoot Uruguay 2011.  
<http://www.youtube.com/watch?v=dmo2Gc9RXBQ>
8. Fontelo P. Extending the benefits of One Laptop per Child to health. BMJ. 2008 Dec 1;337:a2459. doi: 10.1136/bmj.a2459.. Disponible gratuitamente en:  
<http://www.bmj.com/content/337/bmj.a2459.full>
9. Gabriel Kaplún, El ateneo electrónico, CELACOM – ENDICOM - WACC – São Paulo 2004,  
[http://evidoctor.evimed.net/sites/default/files/mailling/imagenes\\_globales/materialPublicaciones/Kaplun-Ateneo\\_electronico.pdf](http://evidoctor.evimed.net/sites/default/files/mailling/imagenes_globales/materialPublicaciones/Kaplun-Ateneo_electronico.pdf)
10. Dr. Alvaro Margolis, Ing. Antonio López. PROCESOS EDUCATIVOS, REDES SOCIALES Y APLICACIONES A LA DIFUSION DE CONOCIMIENTO  
[http://evidoctor.evimed.net/sites/default/files/mailling/imagenes\\_globales/materialPublicaciones/Procesos\\_educativos\\_redes\\_sociales\\_y\\_aplicaciones\\_a\\_la\\_difusion\\_de\\_conocimiento.pdf](http://evidoctor.evimed.net/sites/default/files/mailling/imagenes_globales/materialPublicaciones/Procesos_educativos_redes_sociales_y_aplicaciones_a_la_difusion_de_conocimiento.pdf)
11. Documentación de Moodle: [http://docs.moodle.org/19/es/P%C3%A1gina\\_Principal](http://docs.moodle.org/19/es/P%C3%A1gina_Principal), particularmente la documentación para desarrolladores (Manual de Estilo de Código, Manual de estilo de la interfaz, entro otros).
12. Web oficial del Encuentro MoodleMoot Uruguay: <http://164.73.14.14/moodle/>

13. Sitio oficial Moodle: <http://moodle.org/>
14. GoogleDocs, edición colaborativa de documentos online: <https://docs.google.com>
15. Flickr, sitio para manejar contenido multimedia en la web: <http://www.flickr.com/>
16. YouTube, sitio para manejar contenido multimedia en la web:  
<http://www.youtube.com/>
17. Web oficial de la librería jQuery: <http://jquery.com/>
18. Web oficial de la librería para interfaces de usuario con jQuery: <http://jqueryui.com/>
20. Web de la Librería YUI: <http://developer.yahoo.com/yui/>
21. Eclipse IDE: <http://www.eclipse.org/>
22. Extensión PDT par Eclipse: <http://www.eclipse.org/projects/project.php?id=tools.pdt>
23. MySql: <http://www.mysql.com/>
24. MySql Workbench: <http://www.mysql.com/products/workbench/>
25. Web de XP-DEV: <http://xp-dev.com/>
26. Sitio web de RapidSVN: <http://rapidsvn.tigris.org/>
27. Sitio web de TortoiseSVN: <http://tortoisesvn.tigris.org/>
28. Web de Selenium IDE: <http://seleniumhq.org/projects/ide/>
29. Web de la extensión Firebug para Firefox:  
<https://addons.mozilla.org/es-es/firefox/addon/firebug/>
30. Dropbox: <http://www.dropbox.com>
31. Balsamiq mockups: [www.balsamiq.com](http://www.balsamiq.com)
32. Sitio web de Mantis bug tracker: [www.mantisbt.org](http://www.mantisbt.org)

33. Sitio en Google Sites: <https://sites.google.com/site/educasegen/>

# Capítulo 11 - Anexos

INSERTAR ACA INDICE DE ANEXOS

## Anexo 1 – Manual de Usuario Administrador

A continuación se explican los pasos necesarios para instalar el módulo y sus requerimientos.

### Instalación del módulo

La instalación del módulo se realiza de la forma estándar en Moodle:

1. Descomprimir los archivos en la carpeta */mod* de la instalación Moodle.
2. Dar permisos de escritura a la carpeta *moodle/xml*.
3. Ingresar a Moodle como administrador, ir a *Notificaciones* que se encuentra en la sección *Administración del Sitio* del bloque *Ajustes*, e instalar el modulo.
4. Si se quiere hacer uso de la funcionalidad de visualización de los grafos, deberá tener instalado en el servidor el software Graphviz.

### Instalación de GraphViz

#### Linux:

1. Instalar Pear:

```
sudo apt-get install php-pear (Debian)
```

```
yum install php-pear* (Red Hat)
```

2. Instalar clase Image\_GraphViz ([http://pear.php.net/package/Image\\_GraphViz/](http://pear.php.net/package/Image_GraphViz/))

```
sudo pear install Image_GraphViz-1.3.0 (Debian)
```

```
yum install 'Image_GraphViz-1.3.0' (Red Hat)
```

3. Instalar GraphViz:

```
sudo apt-get install graphviz (Debian)
```

```
yum install 'graphviz*' (Red Hat)
```

4. Reiniciar Apache

```
sudo apache2 -k restart
```

#### Windows:

1. Ir a [www.graphviz.org](http://www.graphviz.org) y descargar el instalador Windows. Una vez

descargado, proceder con la instalación haciendo doble click en el archivo..

2. Luego de instalar GraphViz, instalar Pear (si ya no está instalado): se debe ejecutar el archivo go-pear.bat que se encuentra en el directorio donde está instalado PHP y seguir las indicaciones que se muestran en pantalla.

3. Instalar clases auxiliares de GraphViz para PHP. Abrir una ventana de comandos e ingresar el siguiente comando:

```
pear install Image_GraphViz
```

4. Reiniciar Apache.

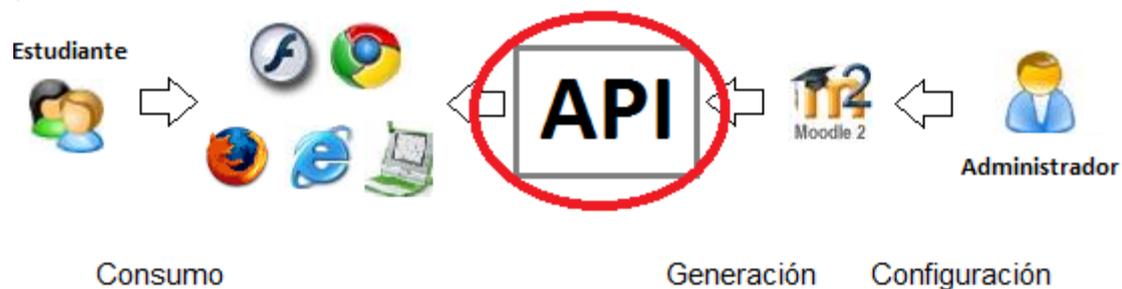
### **Instalación de interfaces para EcgMod**

Para instalar una nueva interfaz, simplemente se debe descomprimir la carpeta con todos los archivos correspondientes a la interfaz en la subcarpeta */mod/ecgmod/themes*.

## Anexo 2– Manual de Usuario Desarrollador

### Biblioteca para Módulo de Consumo (API)

El componente API aísla la capa de presentación de la capa de implementación, brindando una “biblioteca” de funciones que permiten independizar la manera en que se muestra la información.



*Figura - API*

Esta biblioteca expone funciones que permiten manipular libremente la información del caso. Se pueden obtener todos los escenarios y sus datos asociados, las acciones, el estado e historial de los usuarios, salvar el estado y obtener datos estadísticos. Con esta información se pueden crear interfaces totalmente diferentes una de otra y que apunten a diferentes objetivos.

Luego de incluir el archivo `/moodle/mod/ecgmod/api/ecgmod_api.js`, se contará con una serie de estructura de datos y funciones que permitirán manipular la información proveniente del backend. A continuación se detallan dichas estructuras y operaciones.

#### **Estructuras de datos brindadas por la API:**

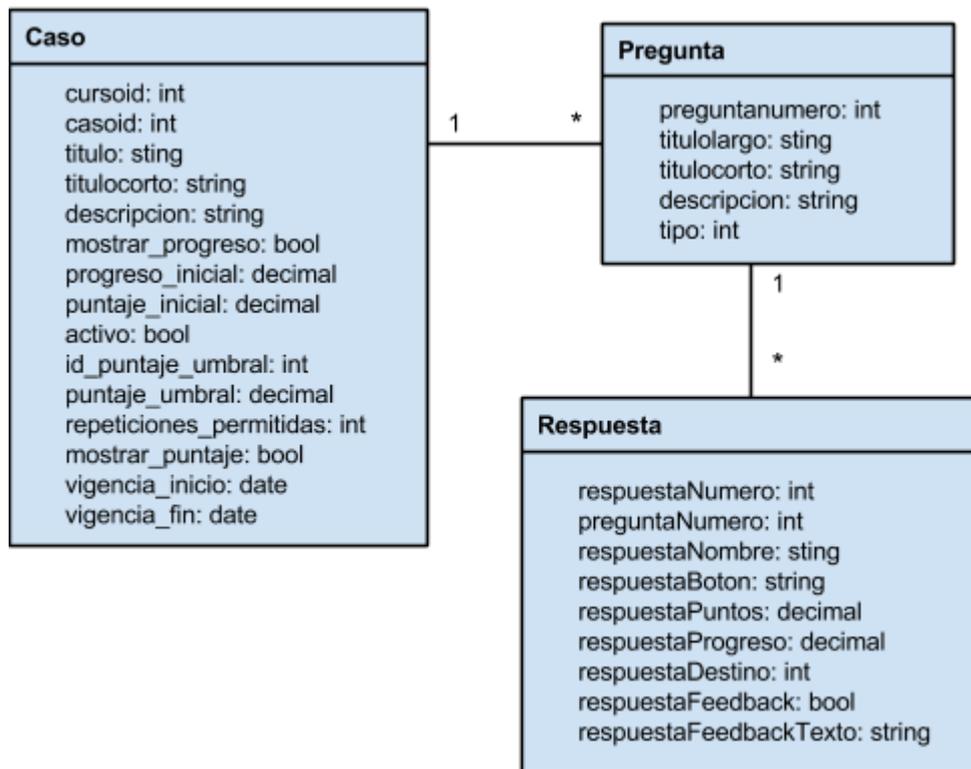


Figura - Estructuras de datos manejados por la API

### Operaciones:

<b>Nombre</b>	getcaso()
<b>Descripción</b>	Retorna el Caso actual
<b>Respuesta</b>	Objeto Caso

<b>Nombre</b>	getpreguntas()
<b>Descripción</b>	Retorna un array de Pregunta (escenarios) con todas los escenarios del caso actual.
<b>Respuesta</b>	Array de objetos Pregunta

<b>Nombre</b>	getpregunta(preguntald)
<b>Descripción</b>	Retorna la Pregunta (escenario) con el Id dado.
<b>Respuesta</b>	Objeto Pregunta

<b>Nombre</b>	getpreguntainicial()
<b>Descripción</b>	Retorna la Pregunta (escenario) inicial del caso.
<b>Respuesta</b>	Objeto Pregunta

<b>Nombre</b>	preguntafinal(preguntald)
<b>Descripción</b>	Retorna true si la pregunta con id dado es final, sino retorna false.
<b>Respuesta</b>	Boolean

<b>Nombre</b>	getrespuestas(preguntald)
<b>Descripción</b>	Retorna todas las Respuesta (acciones) de la pregunta con id dado.
<b>Respuesta</b>	Retorna un array de objetos Respuesta

<b>Nombre</b>	getrespuesta(respuestald)
<b>Descripción</b>	Retorna la Respuesta con id dado.
<b>Respuesta</b>	Objeto Respuesta

<b>Nombre</b>	getrespuestasAll()
---------------	--------------------

<b>Descripción</b>	Retornas todas las respuestas.
<b>Respuesta</b>	Retorna un array de objetos Respuesta

<b>Nombre</b>	sethistoria(cursold, casold, preguntald, respuestald, fecha, puntaje, usuario)
<b>Descripción</b>	Guarda en la base de datos un paso del historial del usuario.
<b>Respuesta</b>	No tiene

<b>Nombre</b>	ecgmod_guardar_estado(cursold, casold, preguntaDestinold, puntaje, progreso, usuario, callback)
<b>Descripción</b>	Guarda en la base de datos el estado del usuario.
<b>Respuesta</b>	No tiene

<b>Nombre</b>	ecgmod_guardar_historia(cursoid, casoid, pregid, respid, puntaje, progreso, usuid, callback)
<b>Descripción</b>	Guarda en el servidor el paso en el historial del usuario.
<b>Respuesta</b>	No tiene

<b>Nombre</b>	ecgmod_cargar(cursold, casold, callback)
<b>Descripción</b>	Carga el caso actual. El parámetro <i>callback</i> es una función que se ejecutará luego de que toda la información del caso haya sido cargada.
<b>Respuesta</b>	No tiene

<b>Nombre</b>	ecgmod_estado_actual(curso_id, caso_id, usuario_id, callback)
<b>Descripción</b>	Carga el estado actual del usuario en el caso El parámetro <i>callback</i> es una función que se ejecutará luego de que toda la información del estado haya sido cargada.
<b>Respuesta</b>	Objeto JSON: <ul style="list-style-type: none"> <li>- curso_id: int</li> <li>- caso_id: int</li> <li>- pregunta_id: int</li> <li>- usuario_id: int</li> <li>- puntaje: decimal</li> <li>- progreso: decimal</li> <li>- timecreated: int</li> <li>- timemodified: int</li> <li>- repeticiones: int</li> </ul>

<b>Nombre</b>	ecgmod_puntaje_promedio(curso_id, caso_id, callback)
<b>Descripción</b>	Retorna el puntaje promedio de todos los usuarios que tomaron el caso El parámetro <i>callback</i> es una función que se ejecutará luego de que toda la información del puntaje haya sido cargada.
<b>Respuesta</b>	Retorna un array de objetos: <ul style="list-style-type: none"> <li>- pregunta_id: int</li> <li>- cantidad: decimal</li> </ul>

<b>Nombre</b>	ecgmod_grafo_recorrido(modulo_id, caso_id, usuario_id)
<b>Descripción</b>	Retorna la ruta a la imagen con el grafo del caso y los pasos del usuario.
<b>Respuesta</b>	String

Las funciones que deben ir contra el servidor (por ejemplo, *ecgmod\_cargar()* o *ecgmod\_guardar\_historia()*), reciben como ultimo parámetro una función anónima que es utilizada como callback cuando se termina de ejecutar la función de la API. Por ejemplo, la función *ecgmod\_cargar()* carga la información del caso de forma asincrónica y luego de finalizada la carga ejecuta el callback pasado como tercer parámetro.

### Manual para desarrollo de interfaces (*themes*)

Para desarrollar una nueva interfaz se deberán cumplir simplemente con dos requisitos:

- encapsular todos los archivos necesarios en una sola carpeta
- contar con un archivo *index.php* que se encarga de renderizar la interfaz

Dentro de la carpeta del *theme* se podrá manejar la estructura de carpetas y archivos que se considere necesaria. A modo de ejemplo, se muestra a continuación la estructura de carpetas para uno de los *themes* desarrollados y que viene por defecto instalado (*tabs*):

```
tabs/
|- css
|- images
|- js
|- index.php
```

Dentro de la carpeta *css* se encuentran todos los archivos de hojas de estilo, en la carpeta *js* los archivos Javascript, en la carpeta *images* las imágenes utilizadas por la interfaz. Además se puede ver el archivo *index.php*, donde está implementada la lógica de la interfaz.

Cuando se ingresa a visualizar un caso, el módulo carga por defecto el archivo *index.php* que se encuentra dentro del directorio de la interfaz y deja disponible para dicho archivo la siguiente estructura de datos almacenada en la variable *\$themeinfo*:

```
$themeinfo->theme: nombre del theme actual
```

```

$themeinfo->course: identificador del curso actual
$themeinfo->caseid: identificador del caso actual
$themeinfo->userid: identificador del usuario logueado
$themeinfo->isgraphenabled: booleano indicando si está habilitado el
modulo de grafos
$themeinfo->moduloid: identificador del módulo
$themeinfo->mode: modo de ejecución, 0 = PREVIEW ; 1 = VIEW (o sea, en
ejecución)
$themeinfo->homeurl: url a la home del curso en Moodle

```

Lo primero que se deberá hacer es cargar la información del caso. Para eso, se deberá ejecutar la función *ecgmod\_cargar()*, función que carga toda la información necesaria para ejecutar el caso. Dicha función realiza la utiliza AJAX para obtener la información desde el servidor (es una carga asincrónica), es por esto que el tercer parámetro de la función es a su vez una función, que será ejecutada al finalizar la carga del caso (es decir, en ese momento se tendrá disponible toda la información necesaria para el caso).

Si se desea recuperar el estado del usuario y retomar el caso desde donde lo había dejado, luego de *ecgmod\_cargar()* debe ejecutarse *ecgmod\_estado\_actual()*. Para que esta funcionalidad cumpla con su cometido es necesario salvar el estado de forma periódica, por ejemplo, cada vez que el usuario realice alguna acción. Para guardar el estado se cuenta con la función *ecgmod\_guardar\_estado()*, la cual permite almacenar los datos del estado actual del usuario: puntaje, avance, última acción tomada, etc.

Luego de ejecutar la función *ecgmod\_cargar()* se tiene disponible la información del caso y se pueden utilizar las diferentes funciones, por ejemplo *getpreguntainicial()*, *getprefunafinal()*, *getrespuestas()*, etc. Utilizando dichas funciones es posible generar la interfaz de la forma que se desee.

Veamos ahora una comparación de la implementación de las interfaces con las que cuenta el módulo por defecto. Simplemente se remarcan las diferencias de implementación entre las interfaces Acordeón y Tabs con el objetivo de conocer a grandes rasgos el esfuerzo necesario para crear una nueva interfaz con un funcionamiento análogo al de las ya existentes.

## Acordeón vs Tabs

### Estilos utilizados

Acordeón: "css/blitzer/jquery-ui-1.8.11.custom.css"

Tabs: "css/smoothness/jquery-ui-1.8.16.custom.css"

### Variables utilizadas

Acordeón: `var _acordeon_preg = [];` // Guarda el índice del Panel para luego

Activar/Desactivar

### Análisis de los cuerpos de las funciones utilizadas con diferencias de implementación

- *function setEstadoBotones(id\_pregunta, disabled)*

- *function ocultarVolver(id\_pregunta)*

acordeón: en los botones se utilizan *classes* y *values* que permiten acceder a cada botón del Escenario para habilitarlo/deshabilitarlo.

Tabs: se habilitan/deshabilitan todos los botones del 'contenedor' del Escenario.

- *function volver()*

Acordeón: se activa explícitamente el 'panel' anterior para que haga el 'SlideDown'.

- *function armarCaso()*

Tabs: activa explícitamente los botones del primer Escenario.

Acordeón: utiliza la etiqueta H3 de HTML como parte del cabezal del 'panel'.

Tabs: utiliza las etiquetas UL y LI de HTML como parte del cabezal del 'panel'.

Acordeón: guarda la información necesaria en '\_acordeon\_preg' para luego activar/desactivar cada 'panel'.

Acordeón: el contenedor de 'paneles' contiene todos los 'paneles', por lo tanto se quita la clase *active* a los paneles que aún no deben estar visibles.

Tabs: el contenedor de 'tabs' contiene solamente el 'tab inicial' y es visible.

- *function addTab(id\_pregunta)*

Acordeón: se utiliza la 'clase' del cabezal del 'panel' para mostrarlo y luego se activa (basandose en '\_acordeon\_preg') para que realice el SlideDown.

Tabs: se agrega el 'tab' al 'contenedor' y luego se activa (basandose en '\_tab\_counter') para que "tome foco".

## HTML

Por tratarse de modelos de interfaz diferentes, existen diferencias en algunas propiedades, por ejemplo en la propiedad *Class*.

# Anexo 3– Manual de Usuario

## Usuario Docente - Configuración de Casos

En este manual se detallan los pasos a seguir para utilizar eficazmente el módulo de generación de ejercicios interactivos integrado a la plataforma Moodle, el cual permite crear dichos ejercicios como un tipo más de Actividad. La herramienta puede ser utilizada por cualquier tipo de curso independiente del tema del mismo.

Los ejercicios interactivos consisten en varios escenarios (estados) posibles y un conjunto de acciones (decisiones) que pueden ser tomadas frente a cada situación planteada.

Las descripciones (tanto de las situaciones como decisiones posibles) permiten cualquier tipo de contenido que pueda ser presentado en una página web (textos, imágenes, audio, videos, applets, etc).

Las transiciones tienen asociadas 'pesos' que permiten medir la calidad de la decisión tomada. Dichos 'pesos' se utilizan para la medición de la *performance* del estudiante durante la ejecución del ejercicio.

Mientras el alumno esté realizando el ejercicio puede avanzar y retroceder, lo que le permite cambiar decisiones anteriores y mejorar su puntaje aprendiendo del ejercicio.

### **Agregar Actividad ‘Caso interactivo (ecgmod)’**

Luego de que el Administrador de la plataforma Moodle haya instalado de forma exitosa el Módulo ‘Caso Interactivo (ecgmod)’, contaremos con ‘ecgmod’ disponible como una Actividad más al momento de elegir entre las posibles Actividades que brinda Moodle.



*Figura - Alta de actividad*

Nota: tener en cuenta que para ‘Agregar actividad...’ es necesario estar en modo Edición activado.

Luego de elegir ‘ecgmod’ en ‘Agregar actividad...’ como la actividad que deseamos agregar, se nos redirige automáticamente a la sección de ajuste de propiedades del módulo.

### **Ajuste de propiedades del Módulo**

En esta sección de configuración es donde se deben ingresar los datos generales de la Actividad que se está agregando. Se muestra a continuación una descripción de los mismos:

- Nombre: es el nombre del Caso, donde es posible agregar una descripción breve pero concisa de la temática del mismo.
- Disponibilidad: permite determinar el período de vigencia de la Actividad que se está agregando, pudiendo detallar con Fecha y Hora el período desde-hasta de la vigencia.
- Tema de la Interfaz: la instalación básica del Módulo ‘ecgmod’ cuenta con dos

opciones disponibles para elegir como la interfaz que será presentada al usuario final (participante del Curso). Es posible contar con otros temas de interfaz desarrolladas por terceros, para conocer más acerca de ello, ver la sección *Manual para desarrollo de interfaces (themes)* en el *Anexo – Manual de Usuario Desarrollador*.

- Ajustes comunes al módulo: soporta la funcionalidad 'Modo de grupo' que brinda Moodle.

**Caso**

Nombre\* Seguridad vial, simplemente un repaso

Disponible desde 15 ▾ October ▾ 2011 ▾ 23 ▾ 00 ▾

Disponible hasta 31 ▾ October ▾ 2011 ▾ 23 ▾ 00 ▾

Tema para la interfaz tabs ▾  
 accordion  
 tabs

**Ajustes comunes del módulo**

Modo de grupo No hay grupos ▾

Agrupamiento\* Ninguno ▾

Visible Mostrar ▾

Número ID

Guardar cambios y regresar al curso Guardar cambios Cancelar

En este formulario hay campos obligatorios\*

Figura - Configuración Caso

Luego de ingresar la información general de la Actividad, al *Guardar cambios* se redirige al usuario automáticamente a la sección de ajuste de propiedades del Caso.

### Ajuste de propiedades del Caso

Esta sección permite definir varios aspectos del Caso que se pueden diferenciar en cuatro tipos:

- Datos generales: Nombre, Nombre abreviado y Descripción. En esta última se puede brindar una introducción al Caso permitiendo contextualizar al usuario final en el mismo.

- Retroalimentación general al usuario

- Progreso Visible: indica si se desea mostrar el Progreso al usuario final. (Progreso general en base a los valores asignados en cada una de las Acciones).

- Puntaje Visible: indica si se desea mostrar el Puntaje al usuario final. (Puntaje general en base a los valores asignados en cada una de las Acciones).

- Reglas sobre la ejecución

- Puntaje Inicial: puntaje con que se inicia el Caso.

- Puntaje Umbral: puntaje límite para forzar la cancelación de la ejecución del Caso. Si no es superado "debería ir a estudiar". Se sugiere que sea menor o igual al Puntaje Inicial.

- Oportunidades Rep.: permite indicar la cantidad de veces que el usuario podrá ejecutar el Caso.

- Información extra: *Interfaz actualizada* indica si la interfaz está actualizada con respecto a las últimas modificaciones que se hayan realizado en alguna de las secciones de configuración.

**Caso**

**Caso\*** Seguridad vial, a modo de ejemplo

**Nombre Abreviado\*** Seguridad vial

Progreso Visible 

Puntaje Visible 

Interfáz actualizada 

**Descripción**

Fuente Tamaño Párrafo    

**B** **I** **U** **ABC** **x<sub>2</sub>** **x<sup>\*</sup>**           

Una intro a las preguntas que hay que saber para obtener la licencia de conducir

Ruta: p

Formato HTML 

Puntaje Inicial  25

Puntaje Umbral  10

Oportunidades Rep.  3 

Guardar cambios Cancelar

Figura - Alta de caso

Luego de configurar todas o algunas de las propiedades mencionadas anteriormente (pues se podrá volver más tarde a esta sección para realizar ajustes), al *Guardar cambios* se nos redirige automáticamente a la vista principal que permite trabajar con cada uno de los componentes de la actividad.

### Vista principal - Trabajar con los distintos componentes de la Actividad

La vista principal sobre la cual se centra la tarea al momento de configurar el Caso (con sus respectivos Escenarios y Acciones) permite tener varias perspectivas del mismo. Se cuenta con una sección que permite editar y visualizar la información que se está ingresando (tab Editar), una sección para previsualizar dicha información permitiendo simular la ejecución del usuario final (tab Vista Previa), y finalmente una sección de reportes estadísticos sobre las ejecuciones de los usuarios (tab Informes). Veamos en detalle cada una de estas perspectivas.

#### Editar

Vista previa   Editar   Informes

Datos del Caso ▶

Estructura del caso ▶

Editar Caso

**Escenarios**

Titulo	Titulo Abreviado	Acciones
Como norma de carácter general (Inicial)	Carácter general	Acciones
Fin (Final)	Fin	Acciones
Vehículos de emergencia	Vehículos de emergencia	Acciones
Uso de bocina	Uso de bocina	Acciones
Cubiertas	Cubiertas	Acciones
Adelantamiento	Adelantamiento	Acciones

Agregar Escenario

Figura - Vista general del caso

- Datos del Caso: se despliegan los datos principales del Caso.

Datos del Caso ▾

**Nombre:** Seguridad vial, a modo de ejemplo

**Descripción**  
Una intro a las preguntas que hay que saber para obtener la licencia de conducir

Figura - Información Caso

- Estructura del Caso: se despliega un grafo a través del cual se modela la relación definida entre los Escenarios y sus respectivas Acciones, siendo cada *nodo* un Escenario y cada *arista* una Acción. Se destaca en color verde el 'Escenario Inicial' y en color gris cada uno de los 'Escenarios Finales'.

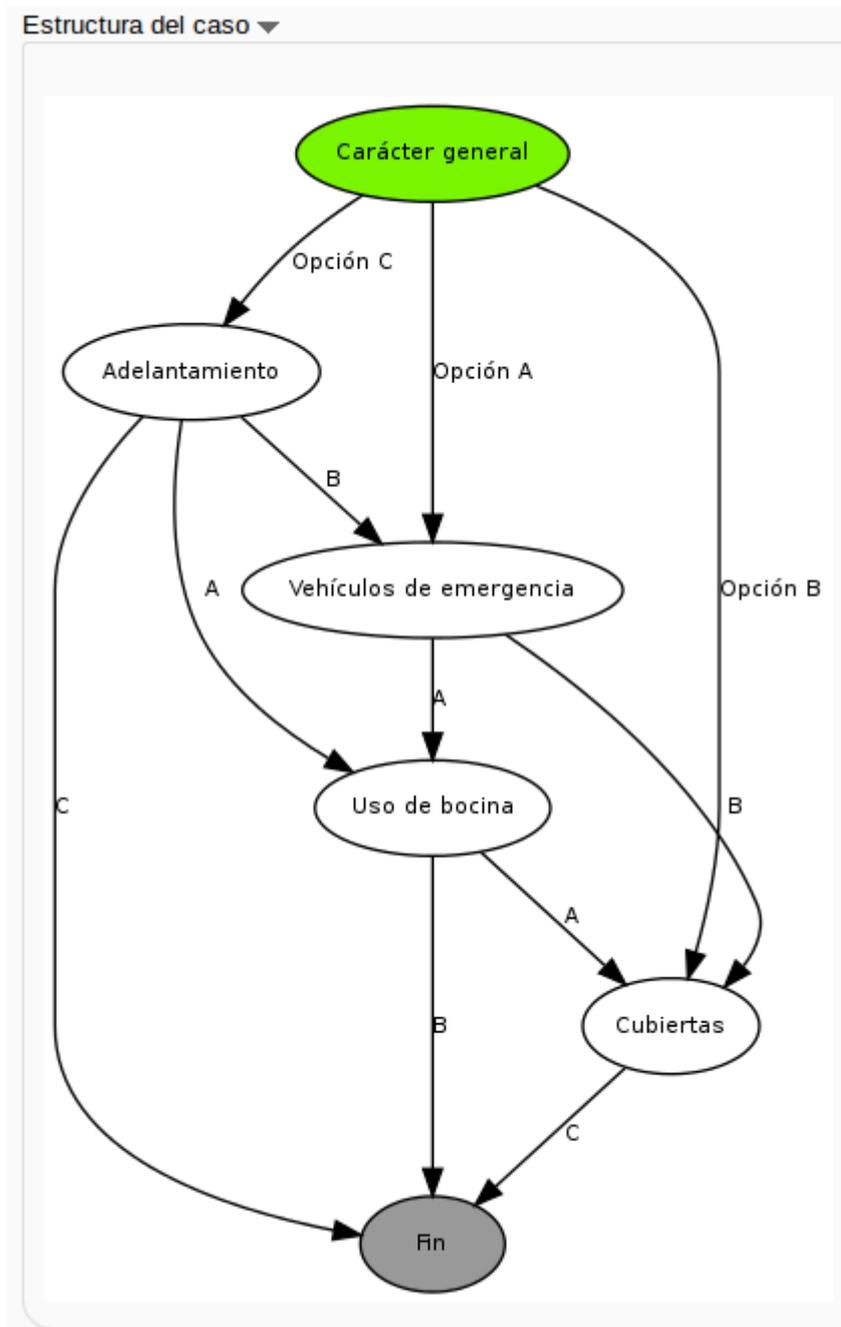


Figura - Grafo del caso

- Editar caso (botón): redirige al formulario de *Ajuste de propiedades del Caso* detallado anteriormente, en donde puede modificarse la configuración actual del Caso.
- Escenarios: se detalla en la siguiente sección.

## Agregar, modificar y eliminar los Escenarios que componen el Caso

Se nos presenta una grilla con el *Título* del Escenario, su *Título Abreviado*, un *link -Acciones-* que redirige a visualizar las Acciones del mismo, y en la última columna dos iconos *-links-* que permiten 'editar'  y 'eliminar'  respectivamente el Escenario en cuestión.

Debajo de la grilla, en el ángulo inferior izquierdo de la misma contamos con un botón *Agregar escenario*, el cual nos redirige al formulario de configuración de un Escenario en donde se deben ingresar los datos correspondientes al mismo.

Si queremos eliminar un Escenario debemos utilizar la opción  como se mencionó anteriormente. Se pedirá una confirmación ya que también serán eliminadas las acciones correspondientes al mismo. El mensaje que solicitara la confirmación es el siguiente:

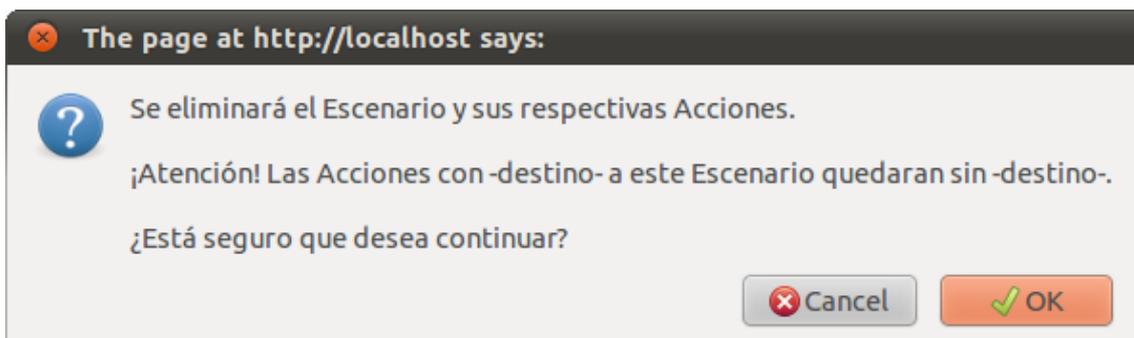


Figura - Eliminación de escenario

Veamos a continuación el formulario de edición de los datos de un Escenario, el cual es utilizado tanto al momento de 'agregar' un nuevo Escenario (botón *Agregar escenario*) como al momento de 'editar'  uno ya existente.

**Escenario**

**Título\*** Vehículos de emergencia

**Título Abreviado\*** Vehículos de emergencia

**Ubicación ?** Intermedia

**Descripción**

Intermedia  
Inicial  
Final

Tamaño Párrafo

Los vehículos de emergencia ¿Cuándo tienen prioridad de paso?



Ruta: p

Formato HTML

Guardar cambios Cancelar

En este formulario hay campos obligatorios\*

Figura - Escenario

Nota: al *Guardar cambios* se retorna a la vista principal del Caso.

Además del *Título* y el *Título abreviado* del Escenario, es importante destacar las propiedades *Descripción* y *Ubicación* del mismo.

- La *Descripción* juega un rol fundamental ya que es donde debe brindarse la información adecuada y plantearse en detalle el contexto del Escenario para que el usuario final pueda tomar la Acción más acertada.
- La *Ubicación* por su parte, permite indicar si el Escenario es el inicial del Caso, alguno de los escenarios intermedios o finales. Se debe tener en cuenta que un Caso debe contar con un único 'escenario inicial', al menos un 'escenario final' y cero o más 'escenarios intermedios'.

En caso de hacer *click* sobre el link *Acciones* correspondiente a cada Escenario (tercer columna en la grilla), se nos redirige a visualizar las Acciones del mismo. Allí se visualizan en una grilla las Acciones posibles del Escenario, pudiéndose 'agregar' nuevas Acciones, 'modificar' o 'eliminar' alguna de las ya existentes.

En la siguiente sección se detalla dicho funcionamiento.

## Agregar, modificar y eliminar las Acciones que componen un Escenario

Se nos presenta una grilla con el título/nombre de la Acción, el *Texto del Botón*, el *Título del Escenario destino* como consecuencia de haber tomado dicha Acción y en la última columna dos iconos *-links-* que permiten 'editar'  y 'eliminar'  respectivamente la Acción en cuestión.

Debajo de la grilla, en el ángulo inferior izquierdo de la misma contamos con un botón *Agregar Acción*, el cual nos redirige al formulario de configuración de una Acción en donde se deben ingresar los datos correspondientes a la misma.

Acción	Texto del Botón	Escenario Destino	
Opción A	Opción A	Vehículos de emergencia	 
Opción B	Opción B	Cubiertas	 
Opción C	Opción C	Adelantamiento	 

Figura - Acción

Si se quiere eliminar una Acción debemos utilizar la opción  y se pedirá una confirmación. El mensaje que solicitara la confirmación es el siguiente:

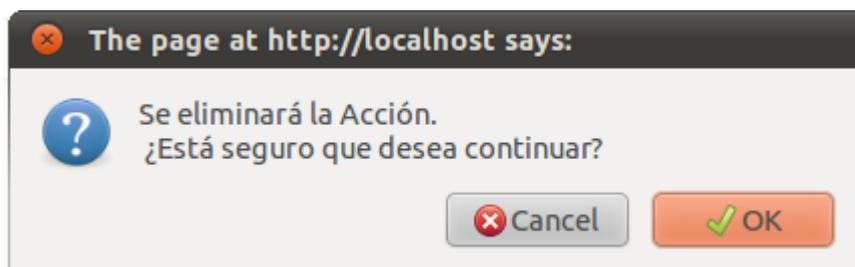


Figura - Eliminar una acción

Veamos a continuación el formulario de edición de los datos de una Acción, el cual es utilizado tanto al momento de 'agregar' una nueva Acción (botón *Agregar Acción*) como al momento de 'editar'  una ya existente.



Acción.

- Descripción: información que se brinda al usuario final con el fin de ayudarlo a tomar la decisión mas adecuada. En caso de optar por dicha Acción, el usuario utilizara el botón que se le presentara con el texto *Texto del Botón*.

Nota: tener en cuenta que *Texto del Botón* o bien puede ser utilizado como tal a nivel de interfaz, o bien como link o como el desarrollador de una nueva interfaz lo desee.

- Post-acción:

- Puntaje: es el puntaje (valor entero) que se suma al puntaje general del usuario en caso de haber tomado dicha Acción. Dicho valor puede ser positivo o negativo, dependiendo de lo que considere adecuado el encargado del Caso.

- Progreso: es el progreso (valor entero con decimales) que se suma al progreso general del usuario en caso de haber tomado dicha Acción. Dicho valor puede ser positivo o negativo, dependiendo de lo que considere adecuado el encargado del Caso.

- Escenario destino: indica cual sera el próximo Escenario que enfrentara el usuario final en caso de que haya optado por esta Acción. Las opciones para elegir el *Escenario destino* son los Escenarios existentes en el Caso, es decir, cualquiera de los presentados en la grilla que se visualiza en la vista general de la perspectiva *Editar*. Se cuenta además con una opción extra "Siguiete escenario", la cual indica que se asignará automáticamente como escenario destino el próximo Escenario que sea agregado al Caso.

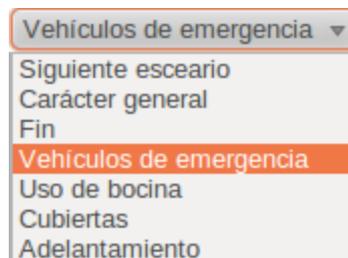


Figura - Próximo escenario

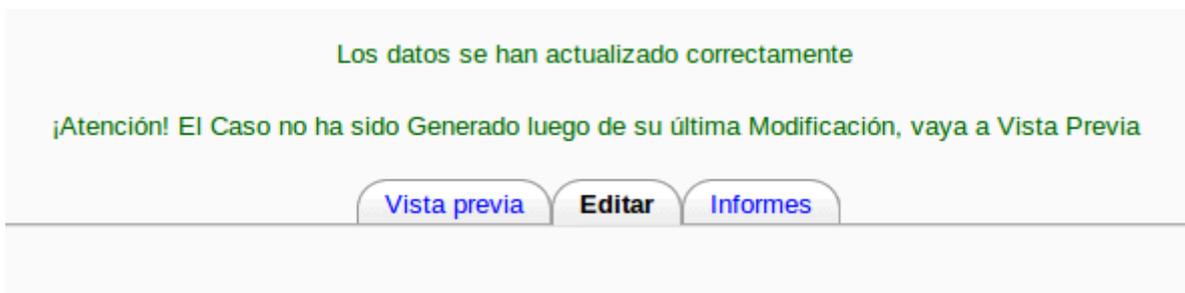
- Mostrar feedback: permite optar entre mostrar o no cierto contenido que brinde una retroalimentación al usuario final luego de haber optado por la Acción. Dicha información sera la configurada en el siguiente ítem.

- Feedback: información que se presentara como retroalimentación al usuario final luego de haber optado por la Acción. Dicha información se mostrara si el ítem *Mostrar feedback* a sido “clickeado”.

Hemos recorrido y configurado cada uno de los ítem necesarios (perspectiva Editar) para que el Caso funcione adecuadamente, por lo tanto, a continuación veremos las dos perspectivas restantes (Vista Previa e Informes).

### Vista Previa

Comencemos un paso antes de ir a la Vista Previa del Caso que hemos configurado. En la siguiente imagen pueden apreciarse dos mensajes, los cuales comentaremos a continuación:



*Figura - Mensajes de edición*

- “Los datos se han actualizado correctamente”: en caso de que hayamos realizado alguna modificación en alguno de los ítems de configuración del Caso (perspectiva Editar), indica que los cambios se han actualizado correctamente.
- “¡Atención! El Caso no ha sido Generado luego de su última Modificación, vaya a Vista Previa”: en caso de que hayamos realizado alguna modificación en alguno de los ítem de configuración del Caso (perspectiva Editar) y aun no hayamos pasado por la Vista Previa, indica que los cambios aun no han sido “reflejados/impactados” en la interfaz que vera el usuario final. Por lo tanto, si quien esta configurando el Caso aun no quiere actualizar las ultimas modificaciones, podria ‘salir’ de la configuración del Caso y el usuario final aun seguiría viendo la versión anterior del mismo (o sea, sin las modificaciones).

En el momento en que el docente va a la Vista Previa, las últimas modificaciones realizadas al Caso son impactadas en la interfaz del usuario final, habiendo “pasado con éxito” un chequeo automático de consistencias. Este chequeo de consistencias controla algunos aspectos importantes del Caso que son necesarios para mantener coherencia y un buen funcionamiento del mismo. A continuación se detallan los chequeos realizados:

- Control de vigencia del Caso: aunque luego de generado el Caso este podría utilizarse fuera de su periodo de vigencia (pues eso sería responsabilidad de la implementación de la interfaz utilizada) se realiza este chequeo considerado “preventivo”. El mensaje desplegado en caso de no pasar el chequeo es: “El periodo de vigencia del Caso esta vencido”.
- Puntaje Inicial VS Puntaje Umbral: para evitar que el usuario final sea “bloqueado” desde un comienzo (aunque esto podría evitarse no realizando el chequeo en la interfaz), se controla la relación entre el *Puntaje Inicial* y el *Puntaje Umbral*. El mensaje desplegado en caso de no pasar el chequeo es: “Puntaje Umbral debe ser menor o igual al puntaje Inicial”.
- Único Escenario Inicial: El mensaje desplegado en caso de no pasar el chequeo es: “Debe existir un único Escenario Inicial”.
- Escenarios Finales sin Acciones: El mensaje desplegado en caso de no pasar el chequeo es: “Los Escenarios Finales no deben tener Acciones posibles”.
- Escenarios no Finales con Acciones: El mensaje desplegado en caso de no pasar el chequeo es: “Todo Escenario Inicial o Intermedio debe tener al menos una Acción”.
- “Siguiendo escenario”: controla que no existan Acciones con *Escenario Destino* “Siguiendo escenario”, pues esto es un valor “imaginario”, ya que no llevaría a ningún Escenario particular. El mensaje desplegado en caso de no pasar el chequeo es: “No deben existir Acciones con destino "Siguiendo escenario" “.
- Sin Escenario Inicial: El mensaje desplegado en caso de no pasar el chequeo

es: “Debe existir uno y solamente un Escenario Inicial”.

- Sin Escenario Final: El mensaje desplegado en caso de no pasar el chequeo es: “Debe existir al menos un Escenario Final”.

Veamos finalmente la Vista Previa.



*Figura - Vista previa*

Esta interfaz es una de las interfaces (Acordeón) con las que se cuenta al momento de ajustar las propiedades del Modulo-Actividad, mas precisamente *Tema de la interfaz*. Mencionaremos la relación entre la información que se muestra en la interfaz y algunas propiedades configuradas en el Caso:

- Puntaje general: gráfica con la suma de los puntajes correspondientes a las Acciones tomadas hasta el momento por el usuario.

- Datos generales del Caso: durante toda la ejecución del Caso, en la parte superior de la pantalla se muestran el *Nombre*, *Nombre abreviado* y *Descripción* del Caso.
- Escenario actual: en la figura anterior el usuario se encuentra en el tercer escenario, 'Vehículos de emergencia', habiendo pasado anteriormente por el escenario 'Adelantamiento' y el Inicial 'Carácter general'. Se nos presenta la *Descripción* correspondiente al Escenario y la *Descripción - botón* de cada una de las posibles *Acciones* a tomar. Notemos que además contamos con un botón *Volver*, el cual permite al usuario volver al Escenario anterior pudiendo tomar otra Acción.

Un dato importante no apreciable en la figura anterior es que el usuario puede “navegar” entre los Escenarios por los que ya ha pasado, pudiendo “recordar” la secuencia de Acciones que ha tomado. Veámoslo en la siguiente figura:

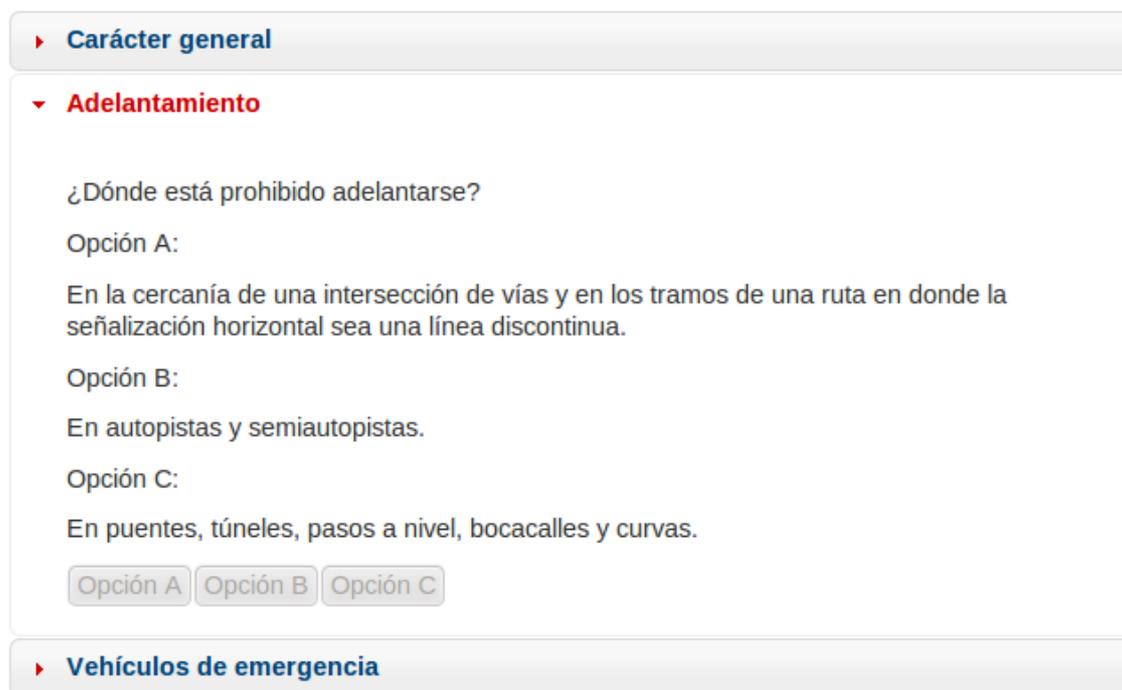


Figura - Recorrer escenarios

Notar que los botones están bloqueados, pues el usuario no ha optado por volver para cambiar la Acción tomada, sino simplemente “recordar” el Escenario anterior.

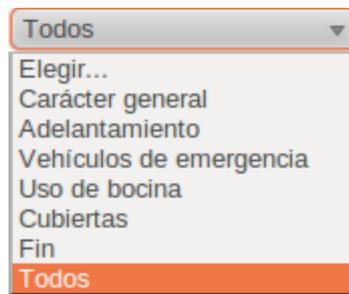
Ya hemos visto una simulación del Caso que ejecutara el usuario. Ahora veremos la perspectiva *Informes* y en la siguiente sección detallaremos algunas otras funcionalidades que son presentadas al usuario final.

## Informes

Escenario	Acción	Cantidad	Porcentaje
Carácter general	Opción A	9	17 %
Carácter general	Opción B	29	55 %
Carácter general	Opción C	15	28 %
Adelantamiento	Opción A	9	69 %
Adelantamiento	Opción B	4	31 %
Adelantamiento	Opción C	0	0 %
Vehículos de emergencia	Opción A	2	40 %
Vehículos de emergencia	Opción B	3	60 %
Uso de bocina	Opción A	2	40 %
Uso de bocina	Opción B	3	60 %
Cubiertas	Opción A	2	22 %
Cubiertas	Opción B	3	33 %
Cubiertas	Opción C	4	44 %

*Figura - Informes*

Ya que el título del informe explica el significado de la información presentada, el único punto que resta destacar es el filtro 'lista de selección desplegable' que en la imagen figura con el valor 'Todos'. Este filtro muestra los escenarios existentes permitiendo filtrar la información presentada por alguno de ellos.

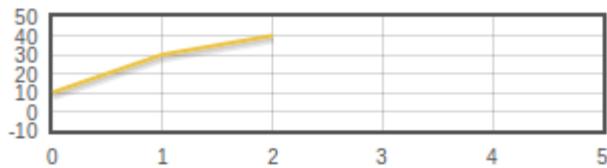


*Figura -Filtro informe*

## **Usuario final, participante del Curso**

En esta sección veremos el Caso que ha sido configurado en la sección “Usuario encargado de un Curso - Configuración de Casos”, destacando además algunas funcionalidades que son presentadas al usuario final pero que no se encuentran disponibles en la Vista Previa.

Retomemos el ejemplo utilizado en la sección mencionada anteriormente.



## Seguridad vial, a modo de ejemplo

### Seguridad vial

Una introducción a las preguntas que hay que saber para obtener la licencia de conducir.

▶ **Carácter general**

▶ **Adelantamiento**

▼ **Vehículos de emergencia**

Los vehículos de emergencia ¿Cuándo tienen prioridad de paso?



Opción A:

Quando están en una situación de emergencia.

Opción B:

Quando están en situación de emergencia y circulan por una avenida.

Opción A Opción B Volver

Si desea salir del Caso para retormarlo luego, haga click [aquí](#)

*Figura - Escenario*

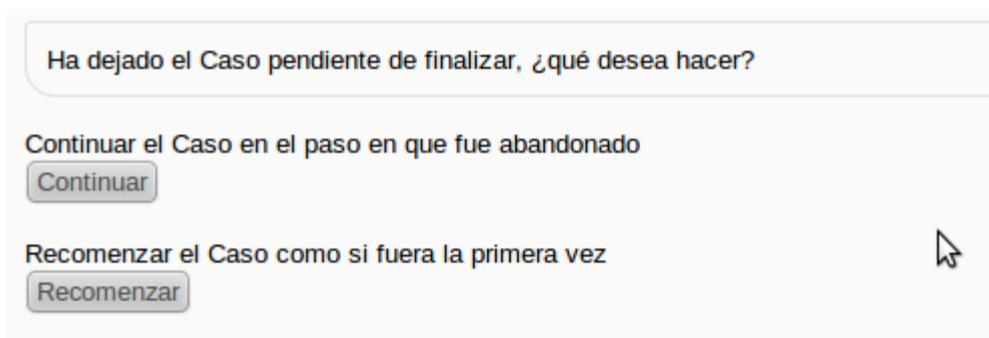
Esta imagen muestra al usuario en el mismo escenario que en el ejemplo anterior, notar dos diferencias:

- No están presentes los tabs de *Vista Previa*, *Editar* e *Informes*, pues están disponibles únicamente para el usuario Docente.

- En la parte inferior de la pantalla (imagen), encontramos un link 'aquí' que permite abandonar el Caso para retormarlo luego. En ese caso, cuando el usuario retome el Caso se le dará la opción de 'continuar' en el Escenario en que abandonó o 'recomenzar'.

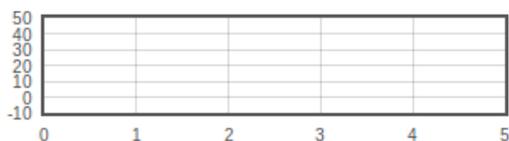
Antes de continuar, optaremos por abandonar el Caso haciendo click en el link 'aquí' que se menciona anteriormente.

Se asume que al otro día se ingresa nuevamente en la Actividad. Se presentará la siguiente situación:



*Figura - Retomar caso*

Si se opta por 'Continuar' y se verá que al momento de retomar el Caso el 'Tema de la interfaz' ha cambiado, ahora se está ante una interfaz de Tabs.



Progreso



## Seguridad vial, a modo de ejemplo

### Seguridad vial

Una introducción a las preguntas que hay que saber para obtener la licencia de conducir.

Vehículos de emergencia

Los vehículos de emergencia ¿Cuándo tienen prioridad de paso?



Opción A:  
Cuando están en una situación de emergencia.

Opción B:  
Cuando están en situación de emergencia y circulan por una avenida.

Opción A Opción B

Si desea salir del Caso para retomararlo luego, haga click [aquí](#)

*Figura - Escenario*

Además, notar que en el ángulo superior derecho apareció una nueva barra que indica el Progreso general en el Caso. Estos dos cambios han sido realizado por el Docente, utilizando las funcionalidades explicadas en la sección “Usuario encargado de un Curso - Configuración de Casos”.

Se toma la opción ‘Opción A’ (*Quando están en una situación de emergencia*).

Vehículos de emergencia    **Uso de bocina**

Con carácter general, ¿está prohibido en zona urbana el uso de bocina en los vehículos?

Opción A:  
Si, está prohibido.

Opción B:  
Está prohibido el uso indebido de la bocina.

Opción A   Opción B   **Volver**

*Figura -Escenario*

Se está en el siguiente Escenario, pero en vez de tomar una de las dos Acciones se opta por 'volver'.

**Respuesta anterior: A**

Vehículos de emergencia

Los vehículos de emergencia ¿Cuándo tienen prioridad de paso?



Opción A:  
Cuando están en una situación de emergencia.

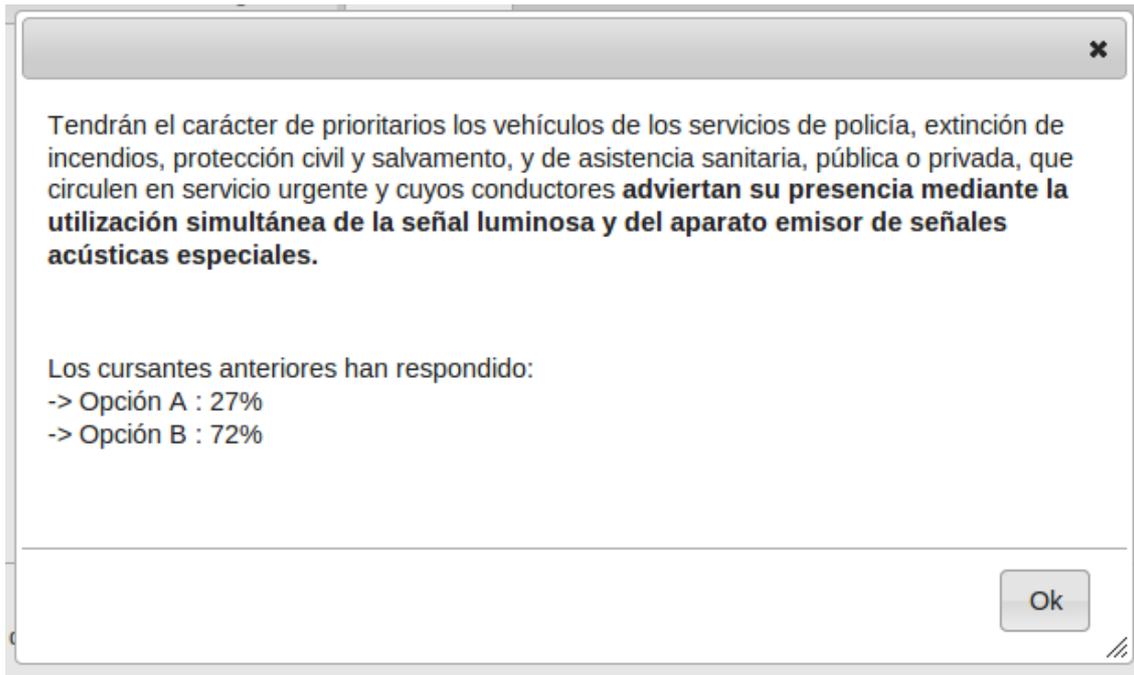
Opción B:  
Cuando están en situación de emergencia y circulan por una avenida.

Opción A   Opción B

*Figura - Escenario volver*

Se vuelve al Escenario anterior, pero notar que ya no se cuenta con el botón 'Volver' y además en la parte superior de los Escenarios se indica cual fue nuestra la anterior.

Ahora se tomará la 'Opción B' (*Cuando están en situación de emergencia y circulan por una avenida*), la cual mostrará un 'pop-up' con la información del *feedback* que fue configurado. Notar que pegado al texto del *feedback* se muestran estadísticas correspondientes a las *acciones* que han tomado los cursantes anteriores.



*Figura - Feedback*

Luego del cerrar el 'pop-up' del feedback se continúa con el Caso hasta que llegamos al final del mismo. Veamos la última pantalla del mismo.

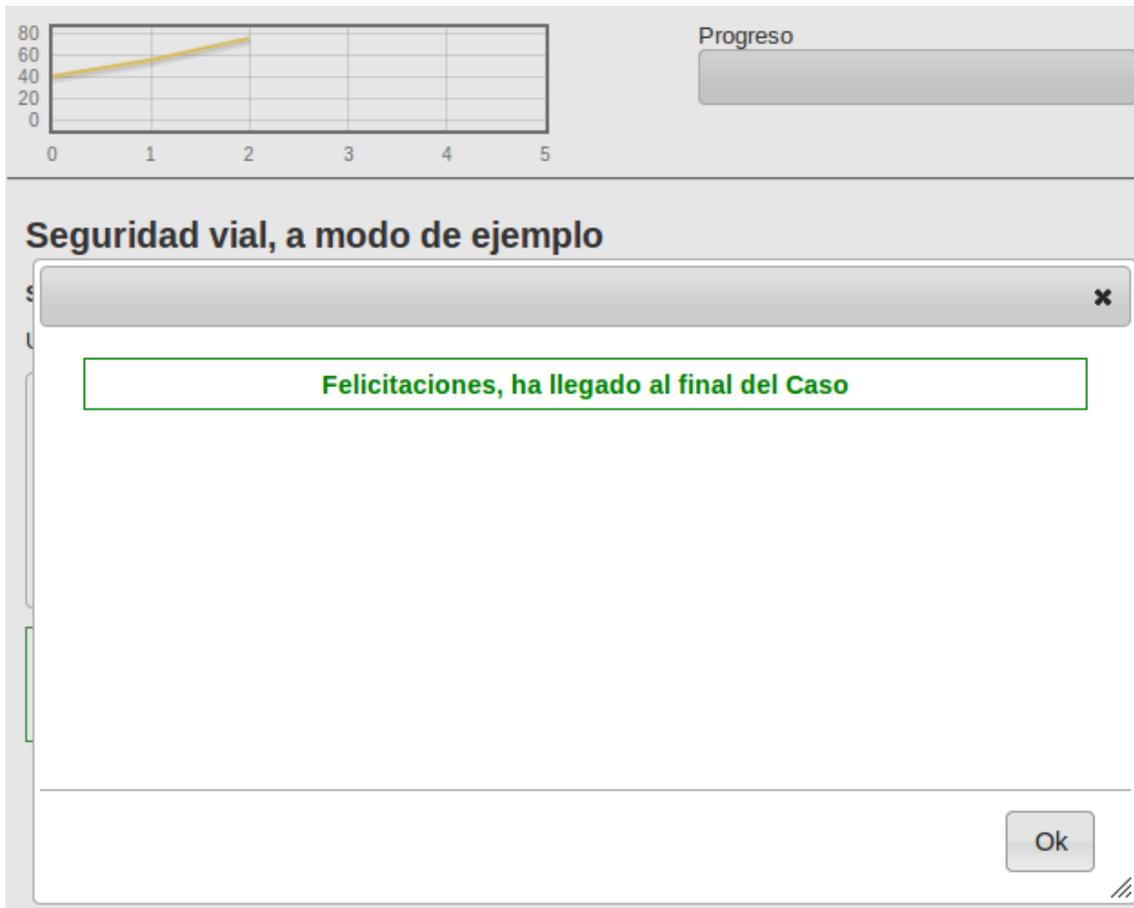


Figura - Final Caso

Se despliega en un 'pop-up' un mensaje que informa que se ha llegado al final del Caso. En este caso, la *Acción* que se tomó no tenía *feedback* para mostrar, por lo tanto únicamente se despliega el mensaje de fin. En caso de que hubiera *feedback* para mostrar, el mismo aparecería en este mismo 'pop-up'.

Luego de cerrar el 'pop-up' se ve el último *Escenario* que simplemente muestra información pero no tiene *Acciones* posibles.

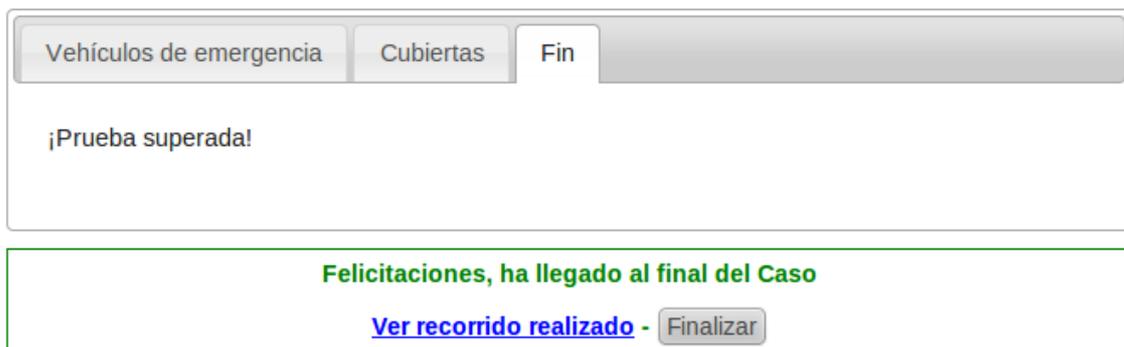


Figura - Fin y recorrido

Antes de *Finalizar*, lo cual llevará al usuario nuevamente al Curso, se puede ver el recorrido que realizado según los *Escenarios* por los que se ha transitado y las

Acciones tomadas. Haciendo 'click' sobre el 'link' *Ver recorrido realizado* se despliega en un 'pop-up' un grafo que modela la relación definida entre los *Escenarios* y sus respectivas *Acciones*, siendo cada *nodo* un Escenario y cada *arista* una Acción.

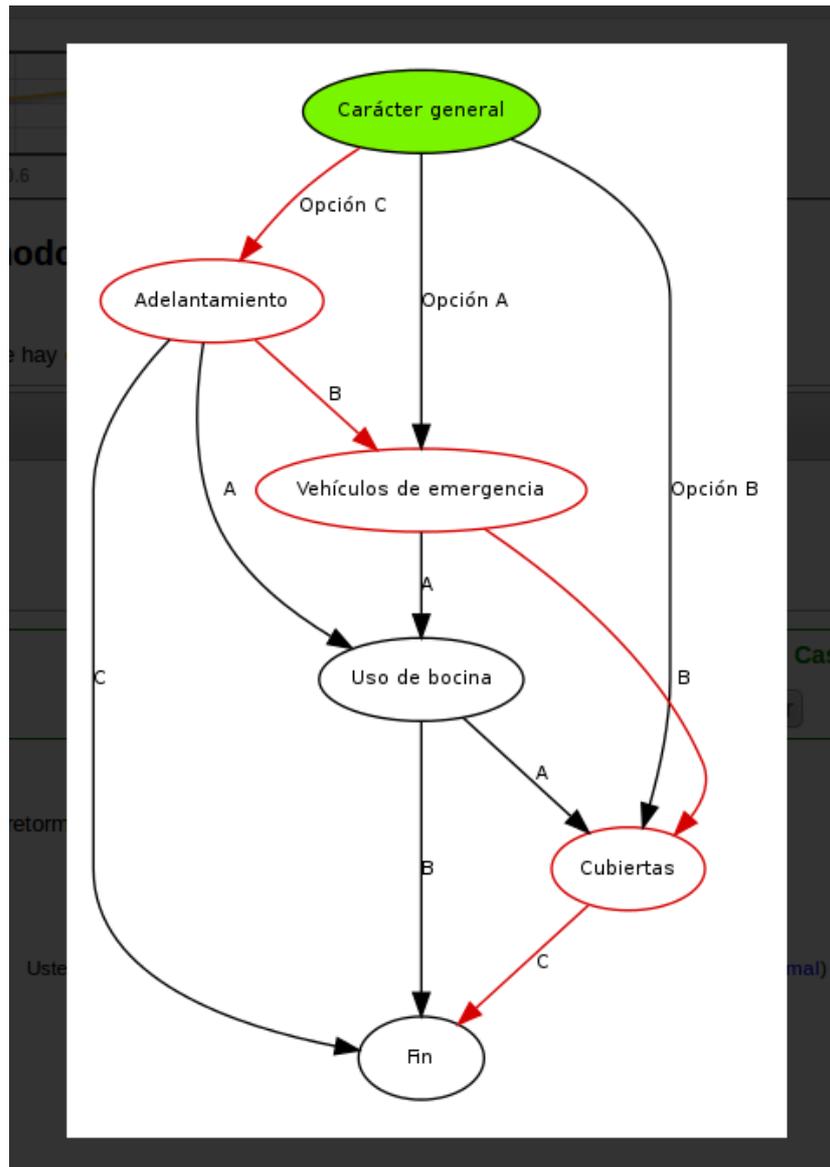


Figura - Recorrido

Se destaca en color verde el 'Escenario Inicial', en color gris cada uno de los 'Escenarios Finales' y en color rojo los *Escenarios* y *Acciones* por los cuales se transitó.

*Observación:* es el mismo grafo que se despliega en la 'Estructura del Caso' de la perspectiva 'Editar' explicada en la sección 'Usuario encargado de un Curso - Configuración de Casos'.

## Anexo 4 - Manual de desarrollo de Módulos en Moodle

Los módulos se almacenan en la carpeta *moodle/mod*, cada uno en un directorio, siendo la estructura general de archivos y directorios la que se describe a continuación:

- *mod\_form.php*: Formulario para crear o modificar una instancia de la actividad.
- *version.php*: para definir meta-información, como por ejemplo la versión del módulo.
- *lang/*: directorio para almacenar los archivos de idioma del módulo. El módulo debe tener archivos de idioma que contenga las traducciones para todos los mensajes y textos del módulo.
- *db/*: Directorio donde se almacenarán los archivos con las tablas de las bases de datos necesarias para la actividad.
- *access.php*: Archivo de permisos del módulo. Los permisos no son obligatorios pero sí muy recomendables para garantizar que usuarios pueden acceder a las distintas partes del módulo.
- *install.xml*: Archivo que describe la estructura de las tablas del módulo.
- *upgrade.php*: código de actualización, aquí es donde se deben de hacer las alteraciones de las tablas, si las hay, entre versiones.
- *index.php*: Este archivo sirve para mostrar todas las instancias de una actividad en un curso, es decir, una lista con todas las instancias del mismo módulo.
- *view.php*: Esta es la página que muestra una instancia de la actividad.
- *lib.php*: biblioteca de funciones del módulo. Aquí se deberán implementar todas las funciones y procedimientos del módulo. Si el módulo se llama “ecgmod”, entonces las funciones obligatorias que deberá tener el módulo tienen que ser de la forma:
  - *ecgmod\_install()*: Acciones a realizar al instalar el módulo.
  - *ecgmod\_add\_instance()*: código para añadir una nueva instancia.
  - *ecgmod\_update\_instance()*: función para actualizar una instancia existente.
  - *ecgmod\_delete\_instance()*: código para borrar una instancia.
  - *ecgmod\_user\_outline()*: da un resumen concreto de la actividad de un usuario.

- `ecgmod_user_complete()`: devuelve un informe más detallado de la contribución de un usuario.
- `ecgmod_get_view_actions()`: Clasifica las acciones para el log. Se usa en el informe de participación.
- Todas las funciones, procedimientos y constantes, creados en `lib.php`, tienen que comenzar con el nombre del módulo.
- `settings.php` (opcional): Formulario con las opciones generales del módulo.

Acerca del archivo `upgrade.php`: en este script se detallan los cambios de base de datos que deben aplicarse en cada instalación. Para cada actualización, se realiza una verificación de la versión:

```
if ($result && $oldversion < 2010091900) {...}
```

y dentro del bloque de código se ejecutan las modificaciones necesarias a la base de datos.

Para la implementación del módulo `EcgMod`, fue necesario agregar nuevos archivos y carpetas, específicos para el módulo:

- `themes/`: En esta carpeta es donde deberán colocarse las diferentes interfaces disponibles para las instancias del módulo (ver [Manual para desarrollo de interfaces \(themes\)](#))
- `api/`: Aquí se tienen los archivos que componen la API.
- `ecgmod_config.php`: Archivo con variables de configuración. Aquí pueden setearse valores para establecer la ubicación de la carpeta de `themes`, si se quiere mostrar el grafo del caso o no, etc.
- `locallib.php`: Archivo que contiene todas las funciones de acceso a datos.
- `preview.php`: Es similar a `view.php`, pero permite previsualizar el caso. Es utilizado al momento de configurar el caso, permitiendo al docente previsualizar cómo va quedando el caso que está creando.
- `viewcasos.php`: Archivo que se encarga de listar todos los casos. En este listado se agregan también los links para iniciar operaciones sobre los casos: crear, editar y eliminar
- `viewpreguntas.php`: Archivo que permite visualizar todos los escenarios del un caso, mostrando además los links para iniciar acciones de crear, editar y eliminar escenarios.
- `viewrespuestas.php`: Archivo que permite visualizar todas las acciones para un

escenario dado e iniciar acciones de crear, editar y eliminar acciones.

- `modedit.php`:
- `grafo_recorrido.php`: Este archivo genera la imagen con el grafo del caso y las aristas correspondientes con el recorrido realizado por el usuario pintadas de diferente color.
- `grafo_img.php`: Este archivo genera la imagen del grafo del caso. Tanto *grafo\_recorrido.php* como *grafo\_img.php* hacen uso de la biblioteca GraphViz para la generación de la imagen del grafo.
- `editcaso.php`: Archivo que procesa la información del caso y almacena los datos en la base de datos.
- `editcaso_form.php`: Archivo que se encarga de generar el formulario de creación y edición de casos.
- `editpregunta.php`: Archivo que procesa la información de un escenario y almacena los datos en la base de datos.
- `editpregunta_form.php`: Archivo que se encarga de generar el formulario de creación y edición de escenarios.
- `editrespuesta.php`: Archivo que procesa la información de una acción y almacena los datos en la base de datos.
- `editrespuesta_form.php`: Archivo que se encarga de generar el formulario de creación y edición de acciones.
- `generarcaso.php`: Este archivo contiene el código de generación del XML del caso. Este script es llamado cuando se genera el caso (actualmente, al hacer el preview el caso es generado automáticamente, pero esto podría ser modificado sin mayores dificultades) y se encarga de crear el archivo XML que luego va a ser consumido por la API. Los archivos son almacenados en la carpeta *moodle/xml*.

## Anexo 5 – Presentación MoodleMootUY

**Artículo enviado para postulación:**

[http://lacio2011.seciu.edu.uy/publicacion/moodlemoot/moodlemootuy2011\\_submission\\_26.pdf](http://lacio2011.seciu.edu.uy/publicacion/moodlemoot/moodlemootuy2011_submission_26.pdf)

**Presentación:**

[http://prezi.com/capjonwloglw/present/?auth\\_key=i98koq7&follow=in5k3gtczo5e](http://prezi.com/capjonwloglw/present/?auth_key=i98koq7&follow=in5k3gtczo5e)

**Videos utilizados para la demo en la presentación:**

Caso Configuracion: <http://www.youtube.com/watch?v=j4rRYmDHbzw>

Caso Presentacion: <http://www.youtube.com/watch?v=dmo2Gc9RXBQ>

## Anexo 6 - Especificación del testing realizado

Inicialmente se realizó un análisis de cuales serían los objetivos de calidad teniendo en cuenta varios aspectos, tales como el producto, las herramientas a utilizar y los distintos tipos de usuarios. En este sentido, algunos de los conceptos manejados fueron:

- calidad respecto a usuario docente: esfuerzo necesario (facilidad de aprendizaje y uso).
- calidad respecto a usuario alumno: frecuencia e impacto de fallas, tiempo de respuesta.
- calidad respecto a usuario administrador: facilidad de instalación y mantenimiento.
- calidad respecto a usuario desarrollador: facilidad de entendimiento del código.

Varios de estos conceptos, tales como *facilidad de uso* y *tiempo de respuesta*, fueron verificados y validados durante el avance del proyecto en las distintas reuniones quincenales con los involucrados de la empresa Evimed.

Respecto a *facilidad de aprendizaje*, *instalación*, *mantenimiento* y *entendimiento del código*, se procuró seguir los lineamientos de Moodle, logrando así un nivel de amigabilidad aceptable a los distintos tipos de usuarios de la plataforma.

En cuanto a la *frecuencia e impacto de fallas* se hizo incapié en el testing durante el desarrollo. Se realizaron diferentes tipos de testing y en momentos distintos dependiendo de la funcionalidad que se quería probar. Se ejecutaron *pruebas unitarias* durante todo el desarrollo y en algunos casos se realizaron *pruebas automatizadas* (scripts) y *pruebas no automatizadas* (revisión entre pares).

- Pruebas unitarias: Se desglosaron las grandes funcionalidades de manera que resultaran en pequeños paquetes de trabajo, logrando así desarrollos de pocas horas que facilitarían el *testing*. Gracias a ello, en la mayoría de los casos fue posible realizar testeo unitario de manera casi inmediata a la implementación.
- Testing automatizado: Se crearon *scripts* para automatizar algunas pruebas

basándose en la técnica de caja negra, básicamente para realizar pruebas de regresión. Es importante destacar que dichas pruebas no fueron ejecutadas de manera independiente al desarrollo, pues el objetivo era obtener una retroalimentación del *testing* de manera ágil e inmediata a la implementación, y por esta razón que no se cuenta con documentación de los resultados de las pruebas.

Respecto a los *tests de configuración del caso*, existen dos *Test Suites*, cada una de ellas con sus *Test Cases*, las cuales se detallan a continuación:

- CaseTS (Configuración de un Caso)
  - TestCase - NewAct\_EditCase: Crea una nueva Actividad en el Curso "actual" y luego edita su contenido.
  - TestCase - EditCase: Simplemente realiza una modificación en el Caso previamente definido.
  - TestCase - AddEscInitial: Agrega el Escenario 'inicial' al Caso definido previamente.
  - TestCase - AddEditDelAction: Agrega y edita una Acción. Luego agrega y elimina otra Acción.
  - TestCase - AddEscInterBack: Agrega un Escenario 'intermedio' que se utilizará solamente para probar la funcionalidad 'volver'.
  - TestCase - AddEscInterFinal: Agrega un Escenario 'intermedio' que se utilizará solamente para llegar al final del Caso.
  - TestCase - AddEscFinal: Agrega el Escenario 'final'.
- PreviewTS
  - TestCase - Preview: Ejecuta en modo Vista-Previa el Caso configurado anteriormente.  
*Observación: hay dos comandos particulares del Test Case que pueden*

*requerir modificaciones antes de ejecutarlo. Luego de cada ejecución de CaseTS es necesario modificar según corresponda los comandos 'click (//button[@value='YY'])' y 'click (//button[@value='ZZ'])'. YY y ZZ son los identificadores que Moodle asignó automáticamente a los Escenarios en cuestión (son los correspondientes al "camino al final").*

En cuanto a los *tests de ejecución del caso*, la *Test Suite* y sus *Test Cases* se detallan a continuación:

- ExecutionTS
  - TestCase - ExeLeaveContinue: Ejecuta en modo *Estudiante* el caso configurado anteriormente. En particular, abandona la ejecución del Caso para retomar luego y *Continuar* con ella.  
*Observación: hay comandos 'click' particulares del Test Case que pueden requerir modificaciones antes de ejecutarlo. Luego de cada ejecución de CaseTS es necesario modificar algún comando según corresponda, por ejemplo 'click (//button[@value='NN'])', pues NN depende del identificador que Moodle haya asignado en la ejecución de CaseTS.*
  - TestCase - ExeLeaveRetake: Ejecuta en modo *Estudiante* el caso configurado anteriormente. En particular, abandona la ejecución del Caso para retomar luego y *Recomenzar* con una nueva ejecución.  
*Observación: hay comandos 'click' particulares del Test Case que pueden requerir modificaciones antes de ejecutarlo. Luego de cada ejecución de CaseTS es necesario modificar algún comando según corresponda, por ejemplo 'click (//button[@value='MM'])', pues MM depende del identificador que Moodle haya asignado en la ejecución de CaseTS.*
- Testing no automatizado: Se realizaron *pruebas de integración* en casos como 'generar y consumir el xml del Caso', 'comunicación de la interfaz con el servidor' y 'controles de integridad al generar el grafo'. Las técnicas utilizadas fueron 'revisión por pares' (verificación estática) y 'ejecución de pruebas'

(verificación dinámica).

Scripts utilizados en el testing automatizado (pruebas de regresión):

- CaseTS (Configuración de un Caso)
  - TestCase - NewAct\_EditCase

NewAct_EditCase		
open	/moodle/course/modedit.php?add=ecgmod&type=&course=2&section=1&return=0	
type	id=id_name	Una actividad de prueba
select	name=availableuntil[year]	label=2012
select	id=id_theme	label=tabs
clickAndWait	id=id_submitbutton	
type	id=id_titulo	Un caso de prueba
type	id=id_titulocorto	Un caso de prueba
click	name=mostrar_progreso	
click	name=mostrar_puntaje	
type	id=tinymce	Descripción del caso para testing
type	id=id_puntaje_inicial	25
type	id=id_puntaje_umbral	5
select	id=id_repeteciones_permitidas	label=10
clickAndWait	id=id_submitbutton	
verifyTextPresent	Los datos se han creado correctamente	

Figura - Script 1

- TestCase - EditCase

EditCase		
clickAndWait	css=div.oneline > form > div > input[type="submit"]	
select	id=id_repeteciones_permitidas	label=99
clickAndWait	id=id_submitbutton	
verifyTextPresent	Los datos se han actualizado correctamente	

Figura - Script 2

- TestCase - AddEsclnitial

AddEscInitial		
clickAndWait	css=div.region-content > div.singlebutton > form > div > input[type="submit"]	
type	id=id_titulo	Primer escenario de prueba
type	id=id_titulocorto	Primer esc. de prueba
type	id=tinymce	Este es el primer escenario de prueba
clickAndWait	id=id_submitbutton	
verifyTextPresent	Los datos se han creado correctamente	

Figura - Script 3

- TestCase - AddEditDelAction

AddEditDelAction		
clickAndWait	link=Acciones	
clickAndWait	css=div.region-content > div.singlebutton > form > div > input[type="submit"]	
type	id=id_nombre	Primer acción del primer escenario de prueba
type	id=id_boton	Opción 1
type	id=tinymce	Esta es la descripción de la primer acción del primer escenario de prueba.
type	id=id_puntos	25
type	id=id_progreso	20
clickAndWait	id=id_submitbutton	
clickAndWait	css=img.iconsmall	
click	name=feedback	
clickAndWait	id=id_submitbutton	
clickAndWait	css=div.region-content > div.singlebutton > form > div > input[type="submit"]	
type	id=id_nombre	Esta acción es el camino para llegar al Fin.
type	id=id_boton	Opción 2
type	id=tinymce	Esta es la segunda acción que se debe tomar luego de la Opción 1.
type	id=id_puntos	50
type	id=id_progreso	100
clickAndWait	id=id_submitbutton	
clickAndWait	css=div.region-content > div.singlebutton > form > div > input[type="submit"]	
type	id=id_nombre	Para eliminar
type	id=id_boton	Eliminar
clickAndWait	id=id_submitbutton	
click	//div[@id='region-main']/div/table/tbody/tr[3]/td[4]/a[2]/img	
assertConfirmation	Se eliminará la Acción. ¿Está seguro que desea continuar?	
waitForAllLinks		
clickAndWait	link=exact:Caso: Un caso de prueba	
verifyTextPresent	¡Atención! El Caso no ha sido Generado luego de su última Modificación, vaya a Vista Previa	

Figura - Script 4

- \* TestCase - AddEscInterBack

AddEscInterBack		
clickAndWait	css=div.region-content > div.singlebutton > form > div > input[type="submit"]	
type	id=id_titulo	Uno de los escenarios intermedios, para testear el Volver.
type	id=id_titulocorto	Un escenario intermedio
type	id=tinymce	Esta es la descripción del escenario intermedio que utilizaremos para probar el funcionamiento del botón Volver.
clickAndWait	id=id_submitbutton	
verifyTextPresent	Los datos se han creado correctamente	
clickAndWait	css=tr.r1 > td.cell.c2 > a	
clickAndWait	css=div.region-content > div.singlebutton > form > div > input[type="submit"]	
type	id=id_nombre	Solamente por cumplir con un botón.
type	id=id_boton	No usar
type	id=tinymce	No optar por esta acción, utilizar el botón Volver para testear su funcionamiento.
clickAndWait	id=id_submitbutton	
verifyTextPresent	Los datos se han creado correctamente	
clickAndWait	link=exact:Caso: Un caso de prueba	
verifyTextPresent	¡Atención! El Caso no ha sido Generado luego de su última Modificación, vaya a Vista Previa	

*Figura - Script 5*

- TestCase - AddEscInterFinal

AddEscInterFinal		
clickAndWait	css=div.region-content > div.singlebutton > form > div > input[type="submit"]	
type	id=id_titulo	El escenario intermedio para llegar al Fin, luego de testear Volver.
type	id=id_titulocorto	Intermedio hacia el Fin
type	id=tinymce	Esta es la descripción del escenario intermedio que utilizaremos para llegar al Fin, luego de probar el funcionamiento del botón Volver.
clickAndWait	id=id_submitbutton	
verifyTextPresent	Los datos se han creado correctamente	
clickAndWait	css=tr.r0.lastrow > td.cell.c2 > a	
clickAndWait	css=div.region-content > div.singlebutton > form > div > input[type="submit"]	
type	id=id_nombre	Para llegar al Fin.
type	id=id_boton	Al Fin
type	id=tinymce	Optar por esta acción, lleva al Fin.
clickAndWait	id=id_submitbutton	
verifyTextPresent	Los datos se han creado correctamente	
clickAndWait	link=exact:Caso: Un caso de prueba	
clickAndWait	link=Acciones	
clickAndWait	//div[@id='region-main']/div/table/tbody/tr[2]/td[4]/a	
select	id=id_pregunta_destino	label=Intermedio hacia el Fin
clickAndWait	id=id_submitbutton	
clickAndWait	link=exact:Caso: Un caso de prueba	

Figura - Script 6

- TestCase - AddEscFinal

AddEscFinal		
clickAndWait	css=div.region-content > div.singlebutton > form > div > input[type="submit"]	
type	id=id_titulo	El escenario final de prueba
type	id=id_titulocorto	El escenario final de prueba
select	id=id_tipo	label=Final
type	id=tinymce	Este es el feedback de la primer acción del primer escenario de prueba.
clickAndWait	id=id_submitbutton	
verifyTextPresent	Los datos se han creado correctamente	

Figura - Script 7

- PreviewTS
  - TestCase - Preview

Preview	
clickAndWait	css=a[title="Vista previa"] > span
verifyTextPresent	Un caso de prueba
verifyTextPresent	Este es el primer escenario de prueba
verifyTextPresent	Opción 1:
verifyTextPresent	Opción 2:
click	css=button.opcion
verifyTextPresent	Los cursantes anteriores han respondido:
click	//button[@type='button']
verifyTextPresent	No usar:
click	css=button[value='volver']
verifyTextPresent	Respuesta anterior:
click	//button[@value='44']
verifyTextPresent	Al Fin:
click	link=Primer esc. de prueba
click	//div[@id='tabs']/ul/li[2]/a/span
click	//button[@value='46']
verifyTextPresent	Felicitaciones, ha llegado al final del Caso
click	//button[@type='button']
verifyTextPresent	Felicitaciones, ha llegado al final del Caso

*Figura - Script 8*

- ExecutionTS
  - TestCase - ExeLeaveContinue

ExeLeaveContinue	
open	/moodle/course/view.php?id=2
clickAndWait	css=#module-38 > div.mod-indent > a > span.instancename
verifyTextPresent	Un caso de prueba
verifyTextPresent	Este es el primer escenario de prueba
verifyTextPresent	Opción 1:
click	css=button.opcion
verifyTextPresent	Los cursantes anteriores han respondido:
click	//button[@type='button']
verifyTextPresent	No usar:
click	//div[@id='tab-55']/button[2]
verifyTextPresent	Respuesta anterior:
verifyTextPresent	Opción 2:
click	//button[@value='44']
verifyTextPresent	Al Fin:
verifyTextPresent	Si desea salir del Caso para retormarlo luego, haga click
clickAndWait	link=aquí
clickAndWait	css=#module-38 > div.mod-indent > a > span.instancename
verifyTextPresent	Continuar el Caso en el paso en que fue abandonado
clickAndWait	css=input[type="submit"]
verifyTextPresent	Al Fin:
click	css=#tabs_58 > button.opcion
verifyTextPresent	Felicitaciones, ha llegado al final del Caso
click	//button[@type='button']
verifyTextPresent	Felicitaciones, ha llegado al final del Caso
verifyTextPresent	Ver recorrido realizado
click	id=vergrafo

Figura - Script 9

- TestCase - ExeLeaveRetake

ExeLeaveRetake	
open	/moodle/course/view.php?id=2
clickAndWait	css=#module-38 > div.mod-indent > a > span.instancename
verifyTextPresent	Un caso de prueba
verifyTextPresent	Este es el primer escenario de prueba
verifyTextPresent	Opción 1:
click	css=button.opcion
verifyTextPresent	Los cursantes anteriores han respondido:
click	//button[@type='button']
verifyTextPresent	No usar:
verifyTextPresent	Si desea salir del Caso para retormarlo luego, haga click
clickAndWait	link=aquí
clickAndWait	css=#module-38 > div.mod-indent > a > span.instancename
verifyTextPresent	Recomenzar el Caso como si fuera la primera vez
clickAndWait	//input[@value='Recomenzar']
verifyTextPresent	Este es el primer escenario de prueba

Figura - Script 10

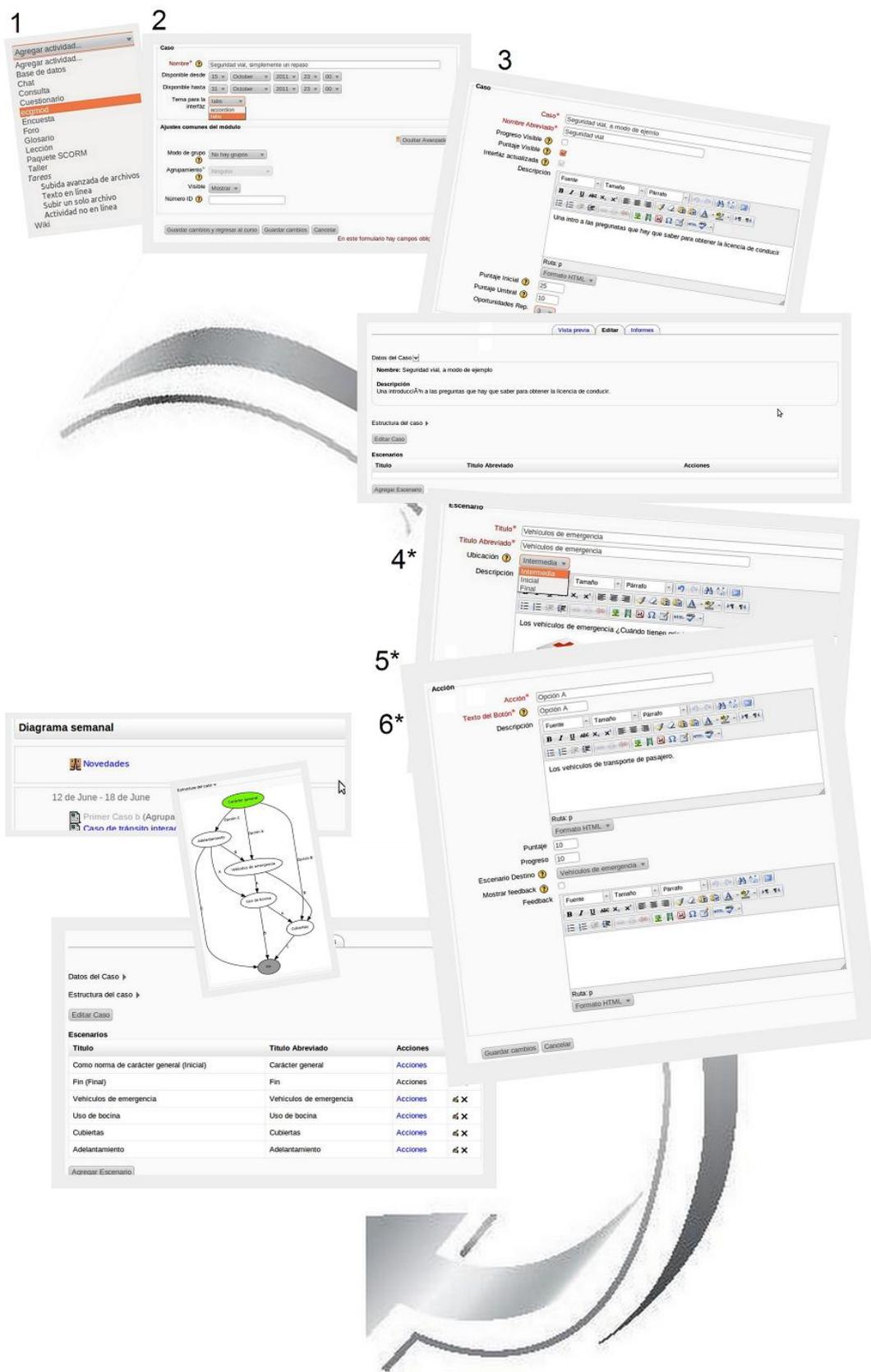


Figura - Recorrido por las distintas secciones de la Actividad