



UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS

PROYECTO DE GRADO

Presentado ante la ilustre UNIVERSIDAD DE LOS ANDES como requisito final para
obtener el Título de INGENIERO DE SISTEMAS

Desarrollo del Sistema de Gestión para la Estrategia RAIS -
GeRAIS Basado en Web Versión 3.0

Por

Br. Yoel Erazo

Tutor: Prof. Beatriz Sandia

Co-tutor: Prof. Gerard Páez

Septiembre 2017

©2017 Universidad de Los Andes. Mérida, Venezuela

Desarrollo del Sistema de Gestión para la Estrategia RAIS – GeRAIS

Basado en Web Version 3.0

Br. Yoel Erazo

Proyecto de Grado — Sistemas Computacionales, 116 páginas

Resumen: La estrategia RAIS (Reproducción de un Ambiente de Innovación en el Salón de clase) propone métodos de enseñanza adaptados al cambio constante de nuestro mundo, de manera tal que, el salón de clases se convierte en un ambiente de producción e innovación en el cual los estudiantes tienen como principal objetivo el desarrollo de un producto. Para esto los estudiantes se organizan en equipos de trabajo o compañías, en las cuales se delegan roles (gerente, director y profesional) y se sigue un plan de trabajo bajo la guía del profesor (director ejecutivo). Actualmente se cuenta con la aplicación web GeRAIS versión 2.0 que da soporte a las tareas y funciones demandadas por la estrategia antes mencionada, por parte de los directores ejecutivos y los equipos de trabajo. En el presente trabajo se busca potenciar y mejorar las funcionalidades presentadas en la versión 2.0 de GeRAIS, generando una versión 3.0 que pueda darle mayor usabilidad a la aplicación facilitando su acceso, distribuyendo los elementos de manera que permitan la comprensión del funcionamiento del sistema, e incorporando nuevos elementos que mejoran sus funcionalidades. La implementación de esta versión 3.0 se ha realizado utilizando el framework back-end Ruby on rails, escrito en ruby, JQuery que es una librería JavaScript, Materialize que es un framework CSS, entre otros.

Palabras Clave: Aplicación web, Educación, Ingeniería Computacional, Plataforma Tecnológica Educativa, Ruby On Rails.

Dedicatoria

A mis padres y hermanos quienes me han acompañado y apoyado en todo momento, y son un ejemplo digno de seguir. Muchas gracias.

www.bdigital.ula.ve

Índice

Dedicatoria.....	iii
Índice.....	iv
Índice de figuras.....	viii
Índice de Tablas.....	x
Capítulo 1 Desarrollo del Sistema de Gestión para la Estrategia RAIS-GeRAIS Basado en Web Versión 3.0.....	1
1.1 Introducción.....	1
1.2 Antecedentes.....	2
1.3 Planteamiento del problema.....	4
1.4 Justificación.....	6
1.5 Objetivos.....	7
1.5.1 Objetivo general.....	7
1.5.2 Objetivos específicos.....	7
1.6 Metodología.....	7
1.6.1 Recolección de historias de los protagonistas.....	8
1.6.2 Análisis de requerimientos y modelo de negocio.....	9
1.6.3 Construcción de prototipos.....	9
1.6.4 Evaluación de prototipo por el cliente.....	10
1.6.5 Refabricación.....	10
1.6.6 Pruebas de la aplicación.....	10
1.7 Alcance.....	11
Capítulo 2 Marco Teórico.....	12
2.1 Especificaciones de la estrategia RAIS.....	12
2.1.1 ¿Qué es la estrategia RAIS?.....	12
2.1.2 Componentes fundamentales de la estrategia RAIS.....	12

2.1.3 ¿Qué es un producto RAIS?.....	13
2.1.4 ¿Cómo se solventan los problemas que surgen de la aplicación de métodos tradicionales de enseñanza a través de la estrategia RAIS?.....	13
2.1.5 ¿Cómo se implementa la estrategia RAIS?.....	14
2.1.6 Reglas de la estrategia RAIS.....	14
2.1.7 Actividades o tareas de la estrategia RAIS.....	15
2.1.8 ¿Cómo se evalúa el desempeño del estudiante en RAIS?.....	15
2.2 Especificaciones técnicas de la herramienta GeRAIS.....	15
2.2.1 Aplicación web.....	15
2.2.2 Arquitectura cliente servidor.....	16
2.2.3 Arquitectura en capas.....	16
2.2.4 Modelado de la base de datos.....	17
Capítulo 3 Requerimientos y modelos.....	18
3.1 Requerimientos del sistema:.....	18
3.1.1 Requerimientos funcionales:.....	18
3.1.2 Requerimientos no funcionales:.....	19
3.1.3 Requerimientos de calidad:.....	19
3.1.4 Requerimientos de seguridad:.....	19
3.2 Modelado del Sistema.....	19
3.2.1 Actores.....	20
3.2.2 Entidades de Negocio.....	21
3.3 Casos de uso.....	22
3.3.1 Actor usuario.....	22
3.3.2 Actor profesor o jefe ejecutivo.....	30
3.3.3 Actor estudiante.....	38
3.3.4 Actor estudiante líder.....	44
3.4 Diagrama de actividades.....	51
Capítulo 4 Diseño y Arquitectura.....	52
4.1 Justificación y diseño.....	52
4.2 Diagrama de despliegue.....	54
4.3 Arquitectura de software.....	54
4.4 Modelo relacional de la base de datos.....	56

4.5 Diagrama de clases.....	58
4.6 Diagrama entidad relación extendido de la base de datos.....	59
4.7 Estructura de la base de datos.....	60
4.7.1 Tabla de comentarios (comments).....	60
4.7.2 Tabla de informe de compromiso (commitment_abstracts).....	60
4.7.3 Tabla de compromiso prototipo (commitment_prototypes).....	60
4.7.4 Tabla de compromisos (commitments).....	61
4.7.5 Tabla de curso-usuarios (course_users).....	61
4.7.6 Tabla de cursos (courses).....	62
4.7.7 Tabla de identificador de Firebase (fcm_token).....	62
4.7.8 Tabla de notificaciones (notifications).....	63
4.7.9 Tabla de posts.....	64
4.7.10 Tabla de informe de productos (product_reports).....	64
4.7.11 Tabla de productos(products).....	64
4.7.8 Tabla de productos prototipos (prototypes).....	65
4.7.8 Tabla de tareas (tasks).....	65
4.7.8 Tabla de informe de tareas.....	66
4.7.8 Tabla de equipos (teams).....	66
4.7.8 Tabla de usuarios (users).....	66
4.8 Diseño de la interfaz de usuario.....	67
Capítulo 5 Especificaciones, desarrollo e implementación.....	70
Desarrollo de GeRAIS versión 3.0.....	70
Capítulo 6 Conclusiones y Recomendaciones.....	74
6.1 Conclusiones.....	74
6.2 Recomendaciones.....	76
Bibliografía.....	78
Apéndice 1 Glosario y definiciones.....	81
8.1 Definiciones técnicas.....	81
8.1.1 HTML.....	81
8.1.2 Navegador web.....	81
8.1.3 Framework.....	82
8.1.4 UML.....	82

8.1.5 MySQL.....	82
8.1.6 Ruby.....	82
8.1.7 Ruby on Rails.....	82
8.1.8 JavaScript.....	83
8.1.9 JQuery.....	83
8.1.10 Firebase.....	83
Apéndice 2 Guías y manuales de uso para el usuario final.....	85
9.1 Instalación de la aplicación.....	85
9.1.1 Pasos de instalación.....	85
9.2 Requisitos para su uso.....	86
9.3 Aspectos generales.....	86
9.4 Guía del usuario general.....	88
9.4.1 Aspecto general.....	88
9.4.2 Editar perfil.....	89
9.4.3 Gestionar notificaciones.....	90
9.4.4 Registrar un nuevo curso.....	91
9.4.5 Solicitar unirse a un curso.....	92
9.4.6 Ingresar al curso a través de código.....	92
9.5 Guía del profesor.....	93
9.5.1 Editar curso.....	93
9.5.2 Eliminar curso.....	93
9.5.3 Editar porcentaje de avance de la tarea.....	94
9.5.4 Editar fecha del compromiso.....	95
9.6 Guía del estudiante.....	95
9.6.1 Retirarse del curso.....	95
9.6.2 Crear un equipo.....	96
9.6.3 Solicitud para unirse a un equipo.....	96
9.6.4 Editar equipo.....	97
9.6.5 Eliminar equipo.....	97
9.6.6 Crear un producto.....	98
9.6.7 Editar un producto.....	98
9.6.8 Eliminar un producto.....	99

9.6.9 Crear compromiso.....	99
9.6.10 Crear informe del producto.....	99
9.6.11 Editar informe del producto.....	100
9.6.12 Editar compromiso.....	100
9.6.13 Eliminar compromiso.....	101
9.6.14 Crear informe del compromiso.....	101
9.6.15 Editar informe del compromiso.....	101
9.6.16 Crear tarea.....	102
9.6.17 Editar tarea.....	102
9.6.18 Eliminar tarea.....	103
9.6.19 Crear informe de tarea.....	103
9.6.20 Editar informe de tarea.....	103
9.6.21 Eliminar informe de tarea.....	103
9.7 Muro del curso.....	105
9.7.1 Crear un comentario.....	105
9.7.2 Editar un comentario.....	105
9.7.1 Editar un comentario.....	105

Índice de figuras

Figura 1: Metodología de desarrollo evolutivo.....	8
Figura 2: Diagrama de actores.....	20
Figura 3: Casos de uso del usuario.....	22
Figura 4: Casos de uso del profesor.....	30
Figura 5: Casos de uso del estudiante.....	38
Figura 6: Casos de uso del estudiante líder.....	44
Figura 7: Diagrama de actividades para registro de compromisos.....	51
Figura 8: Diagrama de despliegue.....	54

Figura 9: Flujo de una aplicación en Ruby on rails.....	55
Figura 10: Diagrama de clases.....	58
Figura 11: Diagrama ERE de la base de datos.....	59
Figura 12: Vista de un curso.....	68
Figura 13: Vista de curso para movil-web.....	69
Figura 14: Pantalla inicial.....	87
Figura 15: Pantalla inicial para móvil y tables.....	87
Figura 16: Pantalla de panel de opciones.....	88
Figura 17: Pantalla de panel de opciones para móvil y table.....	89
Figura 18: Pantalla perfil de usuario.....	90
Figura 19: Pantallas de notificaciones de usuario.....	90
Figura 20: Pantalla para crear curso.....	91
Figura 21: Pantalla para crear curso alternativo.....	91
Figura 22: Pantalla curso solicitud.....	93
Figura 23: Pantalla curso para jefe ejecutivo.....	94
Figura 24: Pantalla tarea para el jefe ejecutivo.....	94
Figura 25: Pantalla compromiso para profesor.....	95
Figura 26: Pantalla curso para estudiante sin equipo.....	96
Figura 27: Pantalla para estudiante que no forma parte de un curso.....	97
Figura 28: Pantalla de equipo para estudiante líder.....	98
Figura 29: Pantalla de producto para estudiante líder.....	100
Figura 30: Pantalla compromiso para estudiante líder.....	102
Figura 31: Pantalla tarea estudiante responsable de la tarea.....	104
Figura 32: Pantalla de tarea para estudiante líder.....	104
Figura 33: Pantalla muro para miembros del curso.....	106

Índice de Tablas

Tabla 1: Tabla de comentarios.....	60
Tabla 2: Tabla de informe de compromiso.....	60
Tabla 3: Tabla de compromisos prototipos.....	61
Tabla 4: Tabla de compromisos.....	61
Tabla 5: Tabla curso-usuarios.....	62
Tabla 6: Tabla de cursos.....	63
Tabla 7: Tabla identificador firebase.....	63
Tabla 8: Tabla de notificaciones.....	64
Tabla 9: Tabla de posts.....	64
Tabla 10: Tabla de informe de productos.....	64
Tabla 11: Tabla de productos.....	65
Tabla 12: Tabla de productos prototipos.....	65
Tabla 13: Tabla de tareas.....	66
Tabla 14: Tabla de informe de la tarea.....	66
Tabla 15: Tabla de equipos.....	67
Tabla 16: Tabla de usuarios.....	67

Capítulo 1

Desarrollo del Sistema de Gestión para la Estrategia RAIS-GeRAIS Basado en Web Versión

3.0

1.1 Introducción.

El enfoque en el mundo académico por lo general no está orientado a un entorno de trabajo lo que trae como consecuencia que los estudiantes no sepan como afrontar la vida laboral. En este sentido la estrategia RAIS proporciona una serie de métodos que tienen como objetivo reproducir un ambiente de trabajo similar al que se observa en un ambiente laboral real, donde además se promueve el emprendimiento y creatividad del estudiante.

Establecer un planteamiento distinto al comúnmente usado da paso a que la evaluación del desempeño académico incorpore nuevos mecanismos, en este caso que permitan evaluar acorde con el trabajo realizado en clases. Es así como la estrategia RAIS plantea hacer seguimiento y evaluación constantes de los estudiantes organizándolos en equipos, los cuales mantendrán reuniones semanales de trabajo e informes semanales de avance del proyecto en ejecución.

Para facilitar llevar a cabo cada uno de los procesos que implica implementar la estrategia RAIS se hace uso de la herramienta tecnológica llamada GeRAIS, que es una aplicación web con arquitectura cliente-servidor, que permite hacer seguimiento continuo del desarrollo y ejecución de un producto, lo que permite evaluar el desempeño de cada uno de los estudiantes. Sin embargo, poder tener una herramienta más dinámica con un nuevo

enfoque que reorganice los ambientes de trabajo es de gran utilidad, pues esto facilita llevar a cabo el proceso de evaluación.

1.2 Antecedentes.

La estrategia de enseñanza aprendizaje RAIS simula las características del ambiente de innovación en el salón de clase, más específicamente, simula el trabajo de lo que se conoce como compañías de arranque, en las que el trabajo efectivo y productivo, es evidente. Entre las características propias de una compañía de arranque se pueden señalar: una estructura organizacional con roles y responsabilidades claramente definidos, donde todos aportan en conocimiento y habilidades; una gran pasión y alto nivel de motivación en los que ejecutan el producto: planificación clara de las actividades a ejecutar y compromisos para lograr las metas y seguimiento y control de los procesos de ejecución del producto (Sandia et al, 2011). Diseñar una asignatura bajo la estrategia RAIS implica incorporar estas características, en la que los estudiantes, agrupados en compañías de arranque, desarrollan un producto, que conduce al logro de los objetivos de aprendizaje y al mejoramiento de las competencias educacionales.

La necesidad de hacer seguimiento y control de los distintos procesos que se dan en la ejecución del producto, y específicamente en los procesos relacionados con el aprendizaje implementando la estrategia RAIS, han generado la necesidad del desarrollo de sistemas que permitan gestionarlos.

Existen en el mercado aplicaciones que permiten hacer seguimiento de los procesos de enseñanza aprendizaje en la web, tales como los Sistemas de Gestión de Aprendizaje (en inglés: *LMS – Learning Management Systems*), entre los que podemos mencionar Moodle (Moodle Trust, 2012) y Docebo (Docebosrl, 2011), los cuales son de código abierto y de uso libre. Estos sistemas consisten un software instalado en un servidor web que permite administrar, distribuir, controlar y hacer seguimiento de las actividades de aprendizaje en línea o electrónico.

Sin embargo, a pesar de que dichas herramientas pueden hacer seguimiento de las actividades de enseñanza aprendizaje ejecutadas, ya sea por estudiantes como profesores, incorporando el uso de TIC, estas no contemplan elementos de simulación de un espacio laboral real y algunas, como Atutor (ATutorCommunity, 2011) que permiten el trabajo colaborativo y cooperativo, no presentan mecanismos para hacer el seguimiento continuo de la ejecución del producto, sus avances y documentación.

Por otro lado, existen también sistemas para hacer seguimiento de proyectos y planes de trabajo que permiten planificar y orientar los procesos del proyecto de principio a fin. Entre estos podemos mencionar Gantt Project, Project HQ, y Team Work, todos de uso libre, los cuales en su mayoría incluyen diagramas de Gantt, y asignación de tareas a las personas que trabajarán en el proyecto, lo que permite hacer seguimiento del proyecto y de los equipos de trabajo, pero no consideran el ambiente de enseñanza aprendizaje. Así mismo, existen otras alternativas de propósito general como por ejemplo Google Grupos (Google, 2012), Grupos Yahoo (Yahoo! Inc, 2012), entre otros; pero tampoco cumplen con la idea de la estrategia RAIS que permita ofrecer una herramienta especializada para facilitar los trabajos de gerencia y seguimiento del desarrollo del producto en el salón de clase por parte de los estudiantes.

El sistema web GeRAIS plantea la gestión de los procesos propios de la estrategia RAIS, permitiendo el seguimiento continuo del desarrollo y ejecución de un producto, así como la consecución de los objetivos educacionales.

A razón de cumplir con lo anteriormente especificado hasta el momento dos versiones enfocadas en la estrategia RAIS han sido desarrolladas, GeRAIS v1.0 y v2.0, siendo estas los antecedentes directos del presente trabajo.

Hocevar (2013), realizó el proyecto de investigación Desarrollo de un sistema de gestión para la estrategia RAIS (GeRAIS). El objetivo principal de dicha investigación fue desarrollar un sistema web, basada en una arquitectura web cliente servidor, que permitiera la gestión, seguimiento y evaluación de la ejecución de uno o varios productos al implementar la estrategia RAIS. En este sentido la herramienta permitía entre otras funciones de micro y macro gerencia, gestionar productos, compañías, integrantes, compromisos,

subcompromisos (tareas individuales), problemas corporativos, autoevaluación, coevaluación, reportes ISA y reportes de calificaciones de la evaluación formativa.

De igual manera Osuna (2015), realizó el proyecto de investigación titulado Desarrollo de la versión 2.0 del sistema para la mejora y seguimiento RAIS (Reproducción de un ambiente de innovación en el salón de clases), la cual tenía por objetivo desarrollar la versión 2.0 de un sistema web, denominado Gestión RAIS (GeRAIS) para la gestión, seguimiento y evaluación de la ejecución de uno o varios productos al implementar la estrategia RAIS (Reproducción de un Ambiente de Innovación en el Salón de Clases) en una asignatura. Con el cual se pretendía la mejora de muchas de las funcionalidades de la versión previa y agregar nuevas funcionalidades, resultando así el rediseño, desarrolló y validación de una nueva herramienta tecnológica automatizada, basada en una arquitectura web cliente servidor, que permitió llevar a cabo las funciones de gerencia, propias del jefe ejecutivo (docente) y propias de un administrador del sistema, implementando la integración y coordinación de múltiples integrantes y múltiples cursos, con canales de comunicación no presenciales (foro).

Es importante destacar que a la par con la elaboración de este trabajo se desarrolla una aplicación móvil para el sistema operativo Android. (Pérez, 2017), en el proyecto de investigación titulado Desarrollo de una aplicación móvil Android para la gestión de productos RAIS (Reproducción de un ambiente de innovación en el salón de clases), tiene como objetivo adaptarse a los nuevos requerimientos que demande la herramienta GeRAIS y además servir de guía, en cuanto a la interfaz de usuario así como gran parte de la funcionalidad y comportamiento, para el desarrollo de este proyecto.

1.3 Planteamiento del problema.

Uno de los principales fundamentos de la estrategia RAIS es la reproducción de la atmósfera interna de trabajo de innovación en el salón de clases. En este sentido organiza a los estudiantes en grupos de trabajos denominados compañías, con una mínima organización jerárquica, las cuales son responsables de la ejecución de un producto que está bajo la supervisión del profesor. La estrategia propone una evaluación formativa, orientada hacia la

ejecución y calidad del producto, para determinar el grado de adquisición de conocimiento, competencias y habilidades del estudiante mediante del seguimiento constante o continuo de su desempeño y la retroalimentación dentro de la compañía por medio reuniones de trabajo e informes de avances semanales, además de una autoevaluación y co-evaluación de cada estudiante.

Con el fin de hacer más fácil la incorporación de la estrategia RAIS en las distintas asignaturas se ha desarrollado una herramienta tecnológica automatizada, llamada GeRAIS, que permite llevar a cabo todas las funciones de micro y macro gerencia. Este sistema web permite gestionar un panel de administración del sistema, panel de cursos, cursos, compañías, integrantes, productos, compromisos, tareas, autoevaluación, co-evaluación, reportes ISA, reportes de calificaciones de la evaluación formativa y permite canales de comunicación entre los miembros a través de un foro.

A pesar de que la herramienta GeRAIS cumple con el manejo e implementación de la estrategia RAIS, dando soporte a las necesidades que se plantea en la estrategia, como ejecutar todo el proceso de macro y micro gerencia de las compañías, hacer y documentar el seguimiento constante de las distintas fases de desarrollo del producto, así como también utilizar el canal de comunicación no presencial (foro), en las experiencias de uso de las versiones previas se ha observado un obstáculo a la hora de crear un nuevo curso así como también la cuenta del encargado de dicho curso ya que dependen de un administrador y no directamente del responsable, en este caso el profesor. De igual manera no permite visualizar todas las herramientas disponibles en una misma pantalla sino que cada elemento en la interfaz de usuario ocupa todo el espacio disponible y en muchos casos solapándose entre sí, además el módulo de comunicación que es un sistema de foro llamado phpbb, que fue integrado a la plataforma, no resulta del agrado del usuario pues comprender como funciona es complicado ya que para crear una conversación o hacer un comentario hay que navegar a través de una serie de enlaces haciendo que el usuario pierda su objetivo. Todas las carencias mencionadas anteriormente derivan en que muchos de los procesos dentro de la herramienta se dificultan por la poca usabilidad presente a la hora de interactuar con esta. Es decir exige un esfuerzo mayor por parte del usuario, para llevar a cabo su ejecución, debido a que no cuenta

con un diseño que permita un fácil acceso a las funcionalidades que implementa y con una adecuada distribución de los elementos para la comprensión del funcionamiento del sistema.

En consecuencia, se requiere el soporte de nuevas funcionalidades, reorganizar y adecuar funcionalidades de los diferentes ambientes de la aplicación que proporcionen mayor usabilidad, mejorando el acceso y la distribución de los elementos, permitiendo así enriquecer el sistema web. De igual manera se requiere la actualización de lenguajes de programación y frameworks utilizados, de forma tal de poder contar con tecnología acorde que permita que la estrategia RAIS llegue a profesores y estudiantes de manera fácil e inmediata promoviendo la incorporación e implementación de la misma.

1.4 Justificación.

La estrategia RAIS permite incorporar nuevos elementos que impulsa una transformación del ambiente académico para la formación de profesionales con conocimientos y habilidades para la solucionar problemas, además de aumentar la motivación para que estos sean proactivos. A través del desarrollo de un producto, que involucra el manejo de conocimiento y el trabajo en grupo, la estrategia RAIS plantea la necesidad de hacer seguimiento y evaluación constante del desempeño de cada compañía y de los estudiantes a través de las reuniones semanales de trabajo y de los informes semanales de avance.

La evaluación basada en la ejecución del producto por parte de los estudiantes no tiene respuestas correctas e incorrectas, son niveles en los cuales se evalúa si el estudiante ha tenido éxito en su desempeño. Llevar a cabo este proceso de evaluación y seguimiento son funciones propias del profesor y las realiza tal y como sucede en un ambiente empresarial. Ejecutar dichas funciones requieren de dedicación y trabajo, por lo que contar con una tecnología de apoyo, que permita llevar a cabo la gerencia, el seguimiento continuo del desarrollo y ejecución del producto, evaluar el desempeño de las compañías y de los miembros (estudiantes) de cada una de ellas, de una forma amigable, fácil, rápida y expedita, es significativo pues facilitaría la implementación de la estrategia en el salón de clases.

Es así que para dar solución a lo planteado anteriormente se genera la propuesta de este trabajo para el desarrollo una nueva versión, siguiendo las historias de usuario

recolectadas, de la herramienta tecnológica llamada GeRAIS, basada en la arquitectura cliente servidor.

1.5 Objetivos.

1.5.1 Objetivo general.

Desarrollar la versión 3.0 de un sistema web, denominado Gestión RAIS (GeRAIS) para la gestión, seguimiento y evaluación de la ejecución de uno o varios productos al implementar la estrategia RAIS (Reproducción de un Ambiente de Innovación en el Salón de Clases) en una asignatura.

1.5.2 Objetivos específicos.

1. Estudiar y analizar en qué consiste la estrategia RAIS, cómo se implementa en el salón de clase, y qué herramientas tecnológicas son usadas actualmente.
2. Estudiar y analizar los requerimientos de la estrategia RAIS para llevar a cabo todo el proceso de macro y micro gerencia.
3. Estudiar exhaustivamente la implementación de la versión 2.0 de GeRAIS (Frameworks, módulos y arquitectura).
4. Analizar y especificar los nuevos requerimientos y los requerimientos a reestructurar para la versión a desarrollar.

1.6 Metodología.

Para el diseño y desarrollo de la nueva versión del sistema planteado, GeRAIS, se mantiene el uso de la metodología de desarrollo evolutivo con entrega de prototipos (Pressman, 2005), para ello se seguirán los pasos que a continuación se exponen:

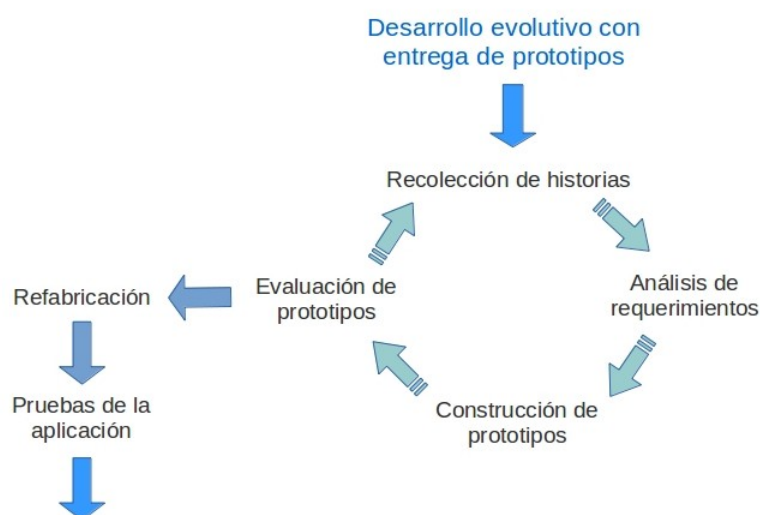


Figura 1: Metodología de desarrollo evolutivo.

1.6.1 Recolección de historias de los protagonistas.

Una historia de usuario es una representación de un requisito de software escrito en una o dos frases utilizando el lenguaje común del usuario. Cada historia de usuario debe ser limitada, esta debería poderse escribir sobre una nota adhesiva pequeña.

Para este proyecto se cuenta con 2 tipos de protagonistas relacionados a la estrategia RAIS, los profesores y los alumnos. Es importante tener presente todas las historias posibles ya que muchas de ellas son aportes útiles para el proyecto y pueden contener posibles soluciones.

En algunos casos, los mismos protagonistas saben cómo se podría solucionar el problema, sin embargo, la solución que proponen puede escapar de sus manos, es ahí donde intervienen los ingenieros. En virtud de esto, las historias de los protagonistas requieren especial cuidado para poder catalogarlas y diferenciar cuáles de ellas tienen sentido, cuáles serán listas de requisitos comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado; y cuáles son reglas de negocio que describen las políticas, normas, operaciones, definiciones y

restricciones presentes en una organización y que son de vital importancia para alcanzar los objetivos.

1.6.2 Análisis de requerimientos y modelo de negocio.

Analizar los requisitos consiste en detectar y corregir las carencias en la comunicación, transformando los requisitos obtenidos de entrevistas previas, en condiciones apropiadas para ser tratados en el diseño. Con base a las historias recolectadas se procede al análisis en donde se determina qué hay que hacer y qué se debe hacer primero, por lo tanto, ya hay algo para trabajar. En este punto es buena práctica sobreestimar los requisitos; esto le dará a la herramienta la posibilidad de adaptarse y crecer durante su desarrollo o en futuras versiones. Con los requisitos más comunes que estén claros, que no haya margen para la duda, se puede trabajar con seguridad; sin embargo, no siempre es así, hay requisitos que son complejos y pueden no estar muy claros. Estos casos se deben trabajar primero con la más alta prioridad y solicitar ayuda a los usuarios mediante el uso de prototipos o versiones previas de la aplicación. Puede que no se tenga claro con certeza hasta donde será el alcance de la aplicación, pero esto no es una razón para detener el desarrollo, al contrario, estimula la creación de prototipos con el fin de extraer con la mayor claridad posible las ideas del usuario.

1.6.3 Construcción de prototipos.

Quizás el punto donde los desarrolladores y los usuarios convergen sea en las interfaces que van a ser usadas por el usuario del sistema. Por esta razón, la metodología propone la creación continua de prototipos que tengan cierta funcionalidad, no importa si estéticamente no son agradables, o si no funcionan en su totalidad. Lo importante es que sean usados para criticar el modelo o la funcionalidad y con la toma de nuevos requisitos o sugerencias ayudar a solucionar las incongruencias que existan. Si el prototipo es bueno, se va a conseguir una buena retroalimentación, que permitirá hacer los arreglos y mejoras al sistema.

1.6.4 Evaluación de prototipo por el cliente.

Mediante el uso del prototipo, el usuario va a opinar y dar sugerencias. Por ejemplo, puede decir que le hace falta cambiar el título, cambiar el tamaño de la letra, colocar una información adicional en un recuadro, entre otras solicitudes. Si la duda fue resuelta en su totalidad se procede a realizar lo mismo con todos los requisitos restantes que estén en la misma situación. Por el contrario, si algunas no fueron resueltas en su totalidad, se debe realizar un nuevo prototipo y repetir el proceso hasta aclarar todas las dudas.

1.6.5 Refabricación.

Luego de realizados todos los pasos antes descritos, hasta el punto de no tener más dudas por aclarar, probablemente no se tenga un sistema estable, pero sí se tendrá una visión clara de la solución y del modelo de negocios. Es decir, en este punto se sabrá cuál es la solución y cómo lograrla, de ésta manera, todo el trabajo previo habrá tenido su fruto y logrado su cometido.

En este momento comienza un proceso de refabricación donde todo se va a asentar en función de la estabilidad y seguridad. Durante este proceso son bienvenidos los cambios y añadidos a petición de los usuarios. En algunos casos, sobre todo en proyectos muy grandes, la refabricación puede verse como una mala práctica. Sin embargo, todo depende del nivel de experticia de los desarrolladores, por supuesto, que en los primeros intentos tendrán sus tropiezos. Por estas razones se recomienda usar esta metodología en proyectos pequeños mientras se adquiere experticia.

1.6.6 Pruebas de la aplicación.

Gracias a la misma naturaleza de la metodología de hacer prototipos funcionales, ya para este punto se cuenta con una buena parte de código probado. Específicamente las pruebas unitarias están adelantadas; faltan pruebas de integración y validación que pueden hacerse en el laboratorio y finalmente se procede junto con los usuarios a hacer las respectivas pruebas

finales del producto. Cabe destacar que se cuenta con la ventaja de que, por lo menos, algún miembro del equipo de usuarios pueda estar entrenado gracias a las versiones previas que el mismo usuario ayudó a evaluar. Por lo tanto, se contará con personal capacitado para estas labores, lo que ahorra tiempo de entrenamiento, vital para colocar con agilidad el producto. La mejor prueba será que el mismo usuario no pueda hacer fallar el sistema (llamada prueba de aceptación), si se logra eso se puede hablar de que el proyecto finalizó exitosamente.

1.7 Alcance.

Basado en la metodología de trabajo, este proyecto tiene planteado el desarrollo de un producto mínimo viable que permita llevar a cabo los procesos propios de la estrategia RAIS. La entrega de prototipos hará posible evaluar si la herramienta cumple con los requerimientos mínimos que posibiliten hacer seguimiento continuo del desarrollo y ejecución de un producto y seguimiento del desempeño de los estudiantes organizados en grupos.

Es importante contar con una herramienta que facilite la implementación de la estrategia RAIS, en la cual los procesos de macro y micro gerencia de una compañía, así como documentación de las distintas fases y el seguimiento continuo de los productos, tareas y estudiantes sean desarrollados en un ambiente de trabajo organizado de manera tal que ofrezca al usuario elementos que permitan resolver las necesidades concretas de la estrategia con un mínimo esfuerzo posible.

Es así como el desarrollo de una nueva versión del sistema web GeRAIS proporciona una herramienta más robusta con una nueva organización, mejorando así muchas de las funcionalidades existentes. Esto brinda la oportunidad de que la estrategia RAIS sea incorporada cada vez más en el salón de clases como método para impartir conocimiento, preparar a los estudiantes para su vida laboral y además de motivarlos a emprender sus propios proyectos.

Capítulo 2

Marco Teórico

2.1 Especificaciones de la estrategia RAIS.

2.1.1 ¿Qué es la estrategia RAIS?

La estrategia RAIS (Reproducción de un Ambiente de Innovación en el Salón de clases) es una estrategia de enseñanza y aprendizaje que propone un esquema de actividades que impulsa la transformación educacional, donde el eje principal en la formación universitaria deja de ser la explicación de conocimientos y pasa a ser el manejo integral de esos conocimientos a través del desarrollo de un producto (Sandia, y otros, 2010). Dicha estrategia plantea incorporar elementos que permitan combatir problemas que surgen a causa del uso de métodos tradicionales de enseñanza como lo son: el bajo nivel de motivación en los estudiantes, el desfase entre avance tecnológico y el proceso de enseñanza aprendizaje, construcción de muros virtuales al conocimiento impartido por el profesor, quien sólo enseña lo que sabe, y el poder formar profesionales que no solo sean capaces de resolver problemas propios de su profesión sino también alcancen un desempeño profesional ético y responsable.

2.1.2 Componentes fundamentales de la estrategia RAIS.

Los componentes de la estrategia RAIS están enmarcados en el contexto de reproducción de un ambiente de innovación en el salón de clases. Existen tres componentes fundamentales:

1. El producto: el cual es el centro de la estrategia RAIS y es mediante su desarrollo que se debe cubrir la esencia del contenido programático de la asignatura. Además debe poder ser realizado con los recursos y talento humano disponible, así como también

- permita una adecuada distribución de tareas y responsabilidades de manera tal que posibiliten el desarrollo de competencias y habilidades del estudiante.
2. Sinergia Capacitación de Conocimientos (SCC): su objetivo es brindar los conocimientos teóricos y prácticos, que el estudiante debe cubrir o necesitará para la elaboración del producto, mediante el uso de métodos y técnicas convencionales como: charlas, discusiones, exposiciones, ejercicios, entre otros.
 3. Sinergia Desarrollo del Producto (SDP): el objetivo principal es proveer el apoyo o ayuda necesaria al estudiante y a su vez hacer seguimiento de su avance durante el periodo de estudio, permitiendo detectar las fallas y carencias que este presente. Esto se logra mediante la simulación de un ambiente empresarial donde se discuten aspectos particulares del producto, se aclaran dudas y resuelven los problemas que surjan.

2.1.3 ¿Qué es un producto RAIS?

Un producto en RAIS es el resultado del desempeño de los estudiantes en el salón de clases y puede ser un prototipo tangible, un modelo, un artículo, una herramienta programada, entre otros. Dicho producto parte de una propuesta del profesor o de un previo acuerdo entre alumno y profesor, lo que permite que en las actividades pautadas para su elaboración se apliquen los conocimientos adquiridos en la asignatura, se verifique la disponibilidad y acceso a los recursos tecnológicos y materiales y pueda ser realizado durante el periodo que dura la asignatura.

2.1.4 ¿Cómo se solventan los problemas que surgen de la aplicación de métodos tradicionales de enseñanza a través de la estrategia RAIS?

Los métodos tradicionales permiten al estudiante adquirir conocimiento, pero muchas veces estos no tienen idea de su utilidad. El hecho de saber que esos conocimientos pueden ser aplicados y evaluados, en un caso real, le facilita al estudiante la apropiación de ese nuevo conocimiento motivando así su deseo por investigar y aprender. Esa motivación da paso al uso de nuevas técnicas y métodos de trabajo más modernos y a su vez actualizando la clase con nuevas ideas (Osuna, 2015). Es así como a través de la elaboración de un producto la estrategia RAIS promueve en los estudiantes realizar de actividades relacionadas con los

conocimientos adquiridos, no conformarse con los conocimientos impartidos por el profesor y aprovechar las herramientas tecnológicas con las que se cuentan hoy en día.

2.1.5 ¿Cómo se implementa la estrategia RAIS?

La estrategia RAIS reproduce dentro del salón de clases algunas de las características de un ambiente real de trabajo, para ello organiza a los estudiantes en grupos de trabajo denominados compañías. Cada uno de los miembros es responsable del desarrollo y ejecución del producto, por tal razón, todos deben trabajar por igual para así de esta forma lograr la calidad y manejo de conocimientos, la disciplina y efectividad de trabajo tal cual una compañía.

Dentro de cada compañía existe una jerarquía la cual permite delegar roles entre los integrantes, tales como:

- Gerente: encargado de administrar el progreso del producto.
- Director: encargado de identificar las necesidades y problemas presentados en la ejecución del producto.
- Profesional: encargado de realizar las tareas semanales.

Sin embargo, todos los miembros juegan el rol de profesionales y están bajo la supervisión del profesor quien desempeña el rol de director ejecutivo.

2.1.6 Reglas de la estrategia RAIS

El desarrollo y ejecución de un producto requiere de reglas y normas que permitan cumplir con los objetivos, en este sentido la RAIS implementa 4 reglas fundamentales las cuales son:

- Cero tolerancia a excusas: promueve el compromiso y la responsabilidad en el estudiante.
- Reunión semanal de trabajo: permite determinar las necesidades o problemas presentados durante el desarrollo del producto, así como también evaluar el desempeño de los estudiantes de manera continua.
- Informe semanal de avance (ISA): este refleja los detalles de las actividades realizadas y a su vez el progreso del producto.

- Administración digital del producto: permite mostrar todo lo relacionado con el progreso del producto por medio de diferentes herramientas tecnológicas.

2.1.7 Actividades o tareas de la estrategia RAIS

Durante el desarrollo del producto es necesario llevar a cabo una serie de actividades o tareas que permitan cumplir con su ejecución. Estas actividades o tareas son planificadas y asignadas semanalmente a cada uno de los miembros la compañía, quienes tienen la obligación de realizar. Cada semana se evalúa la cantidad de actividades cumplidas, las necesidades o problemas presentados en el desarrollo del producto, las actividades que se realizaran y el desempeño de cada estudiante en la actividad de la cual es responsable.

2.1.8 ¿Cómo se evalúa el desempeño del estudiante en RAIS?

La simulación de un ambiente laboral permite hacer un seguimiento constante del desarrollo del producto y así de esta forma garantizar su ejecución. En RAIS el encargado de hacer este seguimiento es el profesor (jefe ejecutivo) quien a lo largo del desarrollo del producto se mantiene en constante vigilancia para verificar el trabajo realizado por la compañía y a su vez evaluar el esfuerzo, aporte y participación de los integrantes. Para llevar a cabo lo antes mencionado y así como determinar el grado de adquisición de conocimientos, competencias y habilidades del estudiante, la estrategia RAIS propone:

1. Una evaluación formativa a través de reuniones semanales de trabajo e informes semanales de avance.
2. Una evaluación sumativa a través de evaluaciones parciales, evaluaciones de avance de ejecución del producto y la presentación final del producto.

2.2 Especificaciones técnicas de la herramienta GeRAIS

2.2.1 Aplicación web

Con respecto al término aplicación web, Pressman(2010) señala:

Llamadas “webapps”, esta categoría de software centrado en redes agrupa una amplia gama de aplicaciones. En su forma más sencilla, las webapps son poco más que un conjunto

de archivos de hipertexto vinculados que presentan información con uso de texto y gráficas limitadas. Sin embargo, desde que surgió Web 2.0, las webapps están evolucionando hacia ambientes de cómputo sofisticados que no sólo proveen características aisladas, funciones de cómputo y contenido para el usuario final, sino que también están integradas con bases de datos corporativas y aplicaciones de negocios.

La evolución de las aplicaciones web han permitido que más usuarios hagan uso de estas herramientas debido a lo práctico de su manejo y las diversas funcionalidades que ofrecen. Hoy día se cuenta con un gran número frameworks que facilitan el desarrollo de las webapps así como documentación compartida por desarrolladores con experiencia.

2.2.2 Arquitectura cliente servidor.

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

2.2.3 Arquitectura en capas.

La estrategia tradicional de utilizar aplicaciones compactas causa gran cantidad de problemas de integración en sistemas complejos como pueden ser los sistemas de gestión o los sistemas de información integrados por más de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad, integración. Para resolver estos problemas se ha generalizado la división de las aplicaciones en capas que normalmente serán tres: una capa que representa la lógica de negocio, permitiéndole al usuario interactuar con el resto de las capas (controladores), una capa para la representación abstracta de los datos (modelo) y por último una interfaz gráfica que facilite al usuario las operaciones sobre los modelos (vistas).

2.2.4 Modelado de la base de datos

El modelo de la base de datos es un tipo de modelo de datos que determina la estructura lógica de una base de datos y de manera fundamental determina el modo de almacenar, organizar y manipular los datos.

Entre los modelos lógicos comunes para bases de datos que se aplican se encuentran:

- Modelo jerárquico
- Modelo en red
- Modelo relacional
- Modelo entidad–relación
- Modelo entidad–relación extendido
- Modelo de objetos
- Modelo documental
- Modelo entidad–atributo–valor
- Modelo en estrella

www.bdigital.ula.ve

Capítulo 3

Requerimientos y modelos.

3.1 Requerimientos del sistema:

El uso de la metodología de desarrollo evolutivo con entrega de prototipos permite estar en comunicación continua con los potenciales usuarios de la herramienta, lo que posibilita la recolección de las historias descritas por los usuarios en lenguaje común y que el ingeniero deberá catalogarlas y diferenciarlas para determinar las necesidades o las condiciones a satisfacer para un software nuevo o modificado.

De todas las historias recolectadas de los protagonistas, las cuales comúnmente pueden ser escritas en pequeñas frases sobre una nota adhesiva se determinó la siguiente lista de requerimientos:

3.1.1 Requerimientos funcionales:

- Gestión de usuarios.
- Gestión de equipos.
- Gestión de productos.
- Gestión de compromisos (asociados permanentemente a productos).
- Gestión de tareas (asociados permanentemente a compromisos).
- Gestión de múltiples cursos
- Elaboración de informes ISA.
- Incorporar canales de comunicación.
- Registro de usuarios en múltiples cursos.

- Registro de usuarios en un equipo del curso.

3.1.2 Requerimientos no funcionales:

- El sistema debe estar disponible en todo momento para el usuario.
- El sistema debe poder ser accedido desde la mayor cantidad posible de dispositivos sin perder funcionalidades.

3.1.3 Requerimientos de calidad:

- El sistema debe ser robusto y estable.
- El sistema debe ser fácil de usar e intuitivo.
- El sistema debe presentar interfaces consistentes conservando el mismo estilo.
- El sistema debe presentar una interfaz, comportamiento y funcionalidad similar a la aplicación móvil Android, desarrollada en la tesis: Desarrollo de una aplicación móvil Android para la gestión de productos RAIS.

3.1.4 Requerimientos de seguridad:

- El sistema debe proporcionar un mecanismo de autenticación de usuarios.
- Proveer mecanismos de protección de la información por posibles errores humanos.

3.2 Modelado del Sistema.

“Un modelo es una representación, en cierto medio, de algo en el mismo u otro medio. El modelo capta los aspectos importantes de lo que estamos modelando, desde cierto punto de vista, y simplifica u omite el resto”(Rumbauch, Jacobson y Booch, 2000, p.XX).

“UML permite modelar sistemas de información, y su objetivo es lograr modelos que, además de describir con cierto grado de formalismo tales sistemas, puedan ser entendidos por los clientes o usuarios de aquello que se modela” (Rumbauch, Jacobson y Booch, 2000, p.11).

“El Lenguaje Unificado de Modelado (UML), que prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan”(Osuna, 2015).

En este sentido, la estrategia RAIS a través del lenguaje UML define dos actores principales, profesor y estudiante.

3.2.1 Actores.

- **Usuario:** puede ser un usuario anónimo que visita el sitio web y tiene la posibilidad de registrarse o ingresar al sistema si se ha registrado previamente, también puede ser un usuario registrado o autenticado una vez haya iniciado sesión el sistema GeRAIS.
- **Profesor:** representa al usuario autenticado que han creado un curso y además de orientar e impartir conocimiento, es el encargado y responsable de evaluaciones y trayectoria del curso. En el sistema GeRAIS, al igual que en la estrategia RAIS se le denomina Jefe Ejecutivo.
- **Estudiante o alumno:** representa al usuario autenticado que forma parte de un curso que no ha sido creado por él. Es el responsable, junto a sus compañeros, de llevar a cabo todas las actividades planificadas en el transcurso de la asignatura. En GeRAIS puede desempeñar el rol de miembro o líder (si ha creado en equipo).

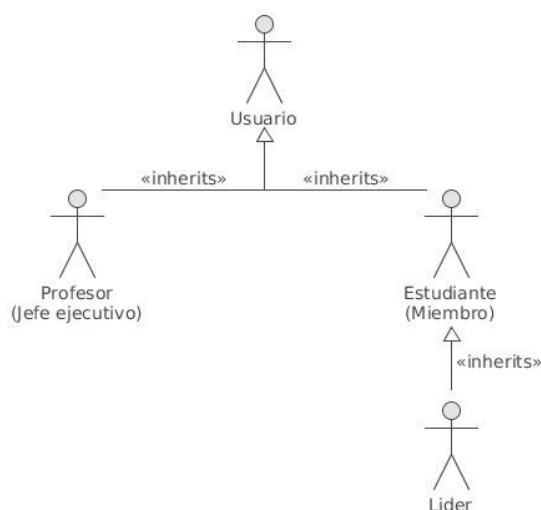


Figura 2: Diagrama de actores

3.2.2 Entidades de Negocio.

En GeRAIS se identifican las siguientes entidades con las cuales los actores pueden interactuar:

- **Producto:** representa el objetivo primordial de la estrategia RAIS, pues es a través de su desarrollo que se pretende impartir los conocimientos relacionados con la asignatura. Puede ser definido por los estudiantes en previo acuerdo con el profesor o partir de una propuesta hecha por el profesor, en tal caso se le denomina producto prototipo.
- **Equipo:** en RAIS se le denomina compañía y relaciona a uno o varios estudiantes con un producto. Cada equipo está conformado por al menos un estudiante, el cual puede desempeñar el rol de miembro o líder (si ha creado el equipo).
- **Compromiso:** es una actividad que está relacionada con un producto y es establecido por el líder del equipo. En caso de tratarse de un producto prototipo este es definido por el profesor y se denomina compromiso prototipo. Tiene una fecha de entrega tope, la cual puede ser pospuesta de ser considerado necesario, así como también un informe de resultados.
- **Tarea:** puede ser una o varias actividades menores que están relacionadas con un compromiso y es asignada por el líder del equipo a cualquier miembro o a sí mismo. Cuenta con un informe de resultados que debe ser entregado antes de la fecha de culminación del compromiso y un porcentaje de avance que es asignado por el profesor (jefe ejecutivo).
- **Informes:** hay tres tipos de informes los cuales engloban resultados que pueden ser comentarios de como se realizó o necesidades que presentan. El primero de ellos es el relacionado con la tarea, editado por el responsable de la tarea y el cual se debe reflejar de manera breve los resultados o posibles problemas, el segundo está relacionado con el compromiso en el cual el líder del equipo refleja de manera resumida el resultado de todas las tareas relacionadas con este y un tercer informe relacionado con el producto que es editado por el líder del equipo y muestra los resultados del mismo.

- **Muro:** es un canal de comunicación donde todos los integrantes de un curso pueden participar, son pequeños posts de 140 caracteres que pueden tener comentarios de 140 caracteres.
- **Curso:** es aquí donde convergen las entidades anteriores por tal razón no pueden existir productos, equipos, muro, compromisos o tareas que no estén asociadas a un curso en particular.
- **Notificaciones:** son alertas que con información que recibirá el usuario para notificarle de alguna actividad.

3.3 Casos de uso.

3.3.1 Actor usuario.



Figura 3: Casos de uso del usuario.

3.3.1.1 Registrar

Nombre: Registrar
Descripción: Permite que un nuevo usuario pueda registrarse en el sistema.
Actores: Usuario.
Precondiciones: ➤ No debe estar registrado en el sistema.
Flujo normal: <u>Crear:</u> ➤ 1. El actor presiona el enlace para registrarse. ➤ 2. El sistema presenta un formulario con los campos que el actor debe llenar. ➤ 3. El actor completa el formulario y presiona el botón de guardar. ➤ 4. El sistema valida los datos de entrada y almacena el registro.
Flujo alternativo: <u>Crear:</u> ➤ 3.A. Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.
Poscondiciones: ➤ El usuario está registrado y entra al sistema.

3.3.1.2 Iniciar sesión

Nombre: Iniciar sesión.
Descripción: Permite al usuario registrado entrar al sistema.
Actores: Usuario.

Precondiciones:
➤ Debe estar registrado en el sistema.
Flujo normal:
<u>Crear:</u>
➤ 1. El sistema presenta un formulario con los campos que el actor debe llenar.
➤ 2. El actor completa el formulario y presiona el botón de guardar.
➤ 3. El sistema valida los datos de entrada y entra al sistema.
Flujo alternativo:
<u>Crear:</u>
➤ 2.A. Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.
Poscondiciones:
➤ El usuario está dentro del sistema.

3.3.1.3 Gestionar curso.

Nombre:
Gestionar curso.
Descripción:
Permite al usuario registrado en el sistema crear un curso con una o varias secciones, editar y eliminar.
Actores:
Usuario.
Precondiciones:
➤ Debe haber iniciado sesión.
Flujo normal:
<u>Crear:</u>
➤ C1. El usuario presiona el enlace para crear un nuevo curso.
➤ C2. El sistema presenta un formulario con los campos que el actor debe llenar
➤ C3. El actor completa el formulario y presiona el botón de guardar.

- C4. El sistema valida los datos de entrada y se crea el curso.

Actualizar:

- A1. El usuario accede al curso en el cual desempeña el rol de jefe ejecutivo.
- A2. El jefe ejecutivo presiona sobre el enlace para editar el curso.
- A3. El sistema muestra el formulario con los campos a modificar.
- A4. El actor edita los campos y presiona el botón de guardar.
- A5. El sistema valida los datos de entrada y se almacenan.

Eliminar:

- E1. El usuario accede al curso en el cual desempeña el rol de jefe ejecutivo.
- E2. El jefe ejecutivo presiona el botón de eliminar.
- E3. El sistema chequea las restricciones y elimina el curso.

Flujo alternativo:

Crear:

- C3.A. Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Actualizar:

- A4.A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Eliminar:

- E2.A El curso no se puede eliminar porque tiene miembros o han sido creados productos prototipos.

Poscondiciones:

Crear:

- Se ha creado un nuevo curso.

Actualizar:

- Se ha actualizado el curso

3.3.1.4 Unirse a un curso.

Nombre: Unirse a un curso.
Descripción: Permite al usuario registrado en el sistema unirse a un curso que ha sido creado por otro miembro del sistema.
Actores: Usuario.
Precondiciones: <ul style="list-style-type: none"> ➤ Debe haber iniciado sesión.
Flujo normal: <u>A través de una solicitud:</u> <ul style="list-style-type: none"> ➤ S1. El usuario presiona el botón para solicitar unirse a un curso. ➤ S2. El jefe ejecutivo del curso acepta la solicitud. ➤ S3. El sistema valida y registra al usuario en el curso. <u>A través de un código:</u> <ul style="list-style-type: none"> ➤ C1. El usuario introduce el código de acceso al curso presiona el botón registrar. ➤ C2. El sistema valida y registra al usuario en el curso.
Flujo alternativo: <u>A través de una solicitud:</u> <ul style="list-style-type: none"> ➤ S1.A. El jefe ejecutivo del curso no acepta o elimina la solicitud. <u>A través de un código:</u> <ul style="list-style-type: none"> ➤ C1.A. El código no coincide con el del curso, se le notifica al usuario.
Poscondiciones: <u>A través de una solicitud:</u> <ul style="list-style-type: none"> ➤ S2. El usuario pasa a ser miembro del curso. ➤ S1.A El usuario aún no forma parte del curso. <u>A través de un código:</u> <ul style="list-style-type: none"> ➤ C2. El usuario pasa a ser miembro del curso. ➤ C1.A. El usuario aún no forma parte del curso.

3.3.1.5 Ver un curso.

Nombre: Ver un curso.
Descripción: Permite al usuario registrado en el sistema ver los cursos de los cuales no forma parte y detalles de los mismos.
Actores: Usuario.
Precondiciones: <ul style="list-style-type: none"> ➤ Debe haber iniciado sesión.
Flujo normal: <u>El curso es público:</u> <ul style="list-style-type: none"> ➤ A1. El usuario presiona enlace del curso y accede ➤ A2. El sistema muestra la descripción del curso, miembros, equipos, productos y muro. ➤ A3. El usuario presiona sobre enlace de algún equipo. ➤ A4. El sistema muestra una descripción, los miembros y productos del equipo. ➤ A5. El usuario presiona sobre enlace de algún producto. ➤ A6. El sistema muestra una descripción, los compromisos actuales y vencidos así como el informe. ➤ A7. El usuario presiona sobre enlace de algún compromiso. ➤ A8. El sistema muestra una descripción, el informe y las tareas asociadas a este. ➤ A9. El usuario presiona sobre alguna tarea. ➤ A10. El sistema muestra una descripción e informe de la tarea. <u>El curso es privado:</u> <ul style="list-style-type: none"> ➤ B1. El usuario presiona enlace del curso y accede ➤ B2. El sistema presenta al usuario una descripción, los miembros y equipos.
Flujo alternativo: <u>El curso es público:</u> <ul style="list-style-type: none"> ➤ A.2.A El sistema no muestra nada en caso de no tener equipos, productos y

comentarios en el muro.
➤ A3.A. El usuario presiona sobre algún producto.(sigue el mismo flujo a partir de A6).
➤ A4.A El sistema no muestra nada en caso de no tener productos.
➤ A6.A El sistema no muestra nada en caso de no tener compromisos e informe.
➤ A8.A El sistema no muestra nada en caso de no tener las tareas e informe.
➤ A10.A El sistema no muestra nada en caso de no tener el informe de la tarea.
<u>El curso es privado:</u>
➤ No hay flujo alternativo.
Poscondiciones:
➤ Muestra detalles del curso.

3.3.1.6 Editar perfil.

Nombre: Editar perfil
Descripción: Permite que un usuario su perfil.
Actores: Usuario.
Precondiciones: ➤ Debe haber iniciado sesión
Flujo normal: <u>Actualizar:</u> ➤ 1. El actor presiona el enlace para editar perfil. ➤ 2. El sistema presenta un formulario con los campos que el actor debe llenar. ➤ 3. El actor completa el formulario y presiona el botón de guardar. ➤ 4. El sistema valida los datos de entrada y almacena el registro.
Flujo alternativo: <u>Actualizar:</u> ➤ 3.A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se

presenta nuevamente el formulario para que corrija.
Poscondiciones: ➤ El perfil de usuario está actualizado.

3.3.1.6 Gestionar notificaciones.

Nombre: Gestionar notificaciones.
Descripción: Permite que un usuario ver, eliminar notificaciones y además aceptar solicitudes si es el caso.
Actores: Usuario.
Precondiciones: ➤ Debe haber iniciado sesión
Flujo normal: Eliminar: <ul style="list-style-type: none"> ➤ 1. El actor presiona el enlace para ver notificaciones. ➤ 2. El sistema presenta las notificaciones. ➤ 3. El actor presiona el enlace para eliminar. ➤ 4. El sistema elimina la notificación.
Flujo alternativo: No hay.
Poscondiciones: Eliminar: <ul style="list-style-type: none"> ➤ Se ha eliminado notificación

3.3.2 Actor profesor o jefe ejecutivo.

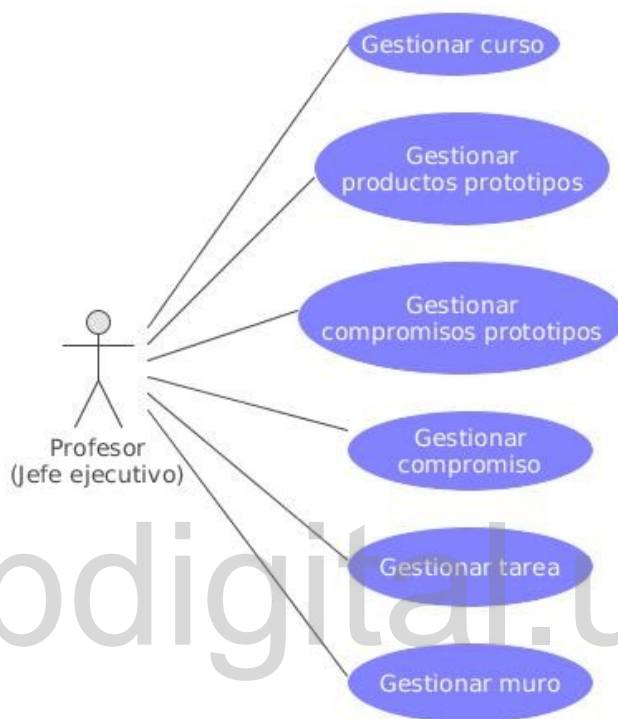


Figura 4: Casos de uso del profesor

3.3.2.1 Gestionar curso

Nombre: Gestionar curso.
Descripción: Permite al usuario registrado en el sistema editar y eliminar un curso creado por él.
Actores: Jefe Ejecutivo o profesor.
Precondiciones: ➤ Debe haber iniciado sesión.

<u>Actualizar:</u> <ul style="list-style-type: none"> ➤ Debe ser el jefe ejecutivo del curso.
Flujo normal: <u>Actualizar:</u> <ul style="list-style-type: none"> ➤ A1. El usuario accede al curso en el cual desempeña el rol de jefe ejecutivo. ➤ A2. El jefe ejecutivo presiona sobre el enlace para editar el curso. ➤ A3. El sistema muestra el formulario con los campos a modificar. ➤ A4. El actor edita los campos y presiona el botón de guardar. ➤ A5. El sistema valida los datos de entrada y guarda los cambios. <u>Eliminar:</u> <ul style="list-style-type: none"> ➤ E1. El usuario accede al curso en el cual desempeña el rol de jefe ejecutivo. ➤ E2. El jefe ejecutivo presiona el botón de eliminar. ➤ E3. El sistema chequea las restricciones y elimina el curso.
Flujo alternativo: <u>Actualizar:</u> <ul style="list-style-type: none"> ➤ A4.A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija. <u>Eliminar:</u> <ul style="list-style-type: none"> ➤ E2.A El curso no se puede eliminar porque tiene miembros o han sido creados productos prototipos.
Poscondiciones: <u>Actualizar:</u> <ul style="list-style-type: none"> ➤ Se ha actualizado el curso. <u>Eliminar:</u> <ul style="list-style-type: none"> ➤ Se ha eliminado el curso.

3.3.2.1 Gestionar productos prototipos.

Nombre: Gestionar productos prototipos.

<p>Descripción:</p> <p>Permite al profesor o jefe ejecutivo crear, editar y eliminar un producto prototipo para el curso.</p>
<p>Actores:</p> <p>Jefe Ejecutivo o profesor.</p>
<p>Precondiciones:</p> <ul style="list-style-type: none"> ➤ Debe haber iniciado sesión. ➤ Debe haber ingresado a un curso donde se desempeña como profesor <p><u>Actualizar:</u></p> <ul style="list-style-type: none"> ➤ Debe ser el jefe ejecutivo del curso.
<p>Flujo normal:</p> <p><u>Crear:</u></p> <ul style="list-style-type: none"> ➤ C1. El profesor presiona sobre la pestaña de productos prototipos y luego sobre el enlace para crear un producto prototipo. ➤ C2. El sistema presenta el formulario que debe llenar. ➤ C3. El actor completa el formulario y presiona el botón de guardar. ➤ C4. El sistema valida y guarda los datos de entrada. <p><u>Actualizar:</u></p> <ul style="list-style-type: none"> ➤ A1. El profesor accede al producto prototipo. ➤ A2. El actor presiona sobre el enlace para editar. ➤ A3. El sistema presenta el formulario con los datos. ➤ A4. El profesor modifica y presiona el botón guardar ➤ A5. El sistema valida y guarda los cambios. <p><u>Eliminar:</u></p> <ul style="list-style-type: none"> ➤ E1. El jefe ejecutivo presiona el botón de eliminar. ➤ E2. El sistema chequea las restricciones y elimina el producto prototipo.
<p>Flujo alternativo:</p> <p><u>Crear:</u></p> <ul style="list-style-type: none"> ➤ C3.A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Actualizar:

- A4.A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Eliminar:

- E2.A El producto prototipo no se puede eliminar porque tiene compromisos prototipos o está siendo desarrollado por algún equipo y se le notifica al profesor.

Poscondiciones:**Crear:**

- Se crea un nuevo producto prototipo.

Actualizar:

- Se ha actualizado el producto prototipo.

Eliminar:

- E2. Se ha eliminado el producto prototipo.
- E2.A El producto prototipo no se ha eliminado

3.3.2.2 Gestionar compromisos prototipos**Nombre:**

Gestionar compromisos prototipos.

Descripción:

Permite al profesor o jefe ejecutivo crear, editar y eliminar un compromiso prototipo para el un producto prototipo del curso.

Actores:

Jefe Ejecutivo o profesor.

Precondiciones:

- Debe haber iniciado sesión.
- Debe haber ingresado a un curso donde se desempeña como profesor
- Debe haber ingresado a un producto prototipo.

Flujo normal:**Crear:**

- C1. El sistema muestra enlace para crear un compromiso prototipo el profesor presiona sobre el.
- C2. El sistema presenta el formulario que debe llenar.
- C3. El actor completa el formulario y presiona el botón de guardar.
- C4. El sistema valida y guarda los datos de entrada.

Actualizar:

- A1. El actor presiona sobre el enlace para editar compromiso prototipo.
- A2. El sistema presenta el formulario con los datos.
- A3. El profesor modifica y presiona el botón guardar
- A4. El sistema valida y guarda los cambios.

Eliminar:

- E1. El jefe ejecutivo presiona el botón de eliminar.
- E2. El sistema chequea las restricciones y elimina el producto prototipo.

Flujo alternativo:

Crear:

- C3.A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Actualizar:

- A3.A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Eliminar:

- E2.A El compromiso prototipo no se puede eliminar porque está siendo desarrollado por algún equipo y se le notifica al profesor.

Poscondiciones:

Crear:

- Se crea un nuevo compromiso prototipo.

Actualizar:

- Se ha actualizado el compromiso prototipo.

Eliminar:

- Se ha eliminado el compromiso prototipo.

3.3.2.3 Gestionar compromisos

Nombre: Gestionar compromisos.
Descripción: Permite al profesor o jefe ejecutivo editar la fecha de un compromiso.
Actores: Jefe Ejecutivo o profesor.
Precondiciones: <ul style="list-style-type: none"> ➤ Debe haber iniciado sesión. ➤ Debe haber ingresado a un curso donde se desempeña como profesor ➤ Debe haber ingresado a un compromiso referente a un producto desarrollado por un equipo.
Flujo normal: <u>Actualizar:</u> <ul style="list-style-type: none"> ➤ A1. El actor presiona sobre el enlace para editar compromiso. ➤ A2. El sistema presenta el formulario con la fecha del compromiso. ➤ A3. El profesor modifica y presiona el botón guardar. ➤ A4. El sistema valida y guarda los cambios.
Flujo alternativo: <u>Actualizar:</u> <ul style="list-style-type: none"> ➤ A3.A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.
Poscondiciones: <u>Actualizar:</u> <ul style="list-style-type: none"> ➤ Se ha actualizado el compromiso.

3.3.2.4 Gestionar tareas

Nombre: Gestionar tareas.

Descripción: Permite al profesor o jefe ejecutivo editar el porcentaje de avance de una tarea.
Actores: Jefe Ejecutivo o profesor.
Precondiciones: <ul style="list-style-type: none"> ➤ Debe haber iniciado sesión. ➤ Debe haber ingresado a un curso donde se desempeña como profesor ➤ Debe haber ingresado a una tarea desarrollada por algún miembro de un equipo del curso.
Flujo normal: <u>Actualizar:</u> <ul style="list-style-type: none"> ➤ A1. El actor presiona sobre la barra de asignación de porcentaje y establece un porcentaje. ➤ A2. El sistema guarda los cambios y comunica al profesor.
Flujo alternativo: <ul style="list-style-type: none"> ➤ No hay flujo alternativo.
Poscondiciones: <u>Actualizar:</u> <ul style="list-style-type: none"> ➤ Se ha actualizado el avance de la tarea.

3.3.2.5 Gestionar muro

Nombre: Gestionar muro.
Descripción: Permite al profesor interactuar con los demás miembros de un curso a través de la creación de pequeños comentarios.
Actores: Profesor.
Precondiciones:

- Debe haber iniciado sesión.
- Debe haber ingresado al curso donde se desempeña como profesor.

Flujo normal:

Crear:

- C1. El sistema presenta un formulario que el profesor debe llenar.
- C2. El profesor edita y pulsa el botón guardar.
- C3. El sistema valida los datos y guarda los cambios.

Actualizar:

- A1. El profesor pulsa el enlace para editar, del comentario que creó.
- A2. El sistema muestra un formulario.
- A3. El profesor modifica y pulsa el botón guardar.
- A4. El sistema valida y guarda los cambios.

Eliminar:

- E1. El profesor presiona el botón eliminar, del comentario que creó.
- E2. El sistema chequea restricciones y elimina el comentario.

Flujo alternativo:

Crear:

- C2..A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Actualizar:

- A3.A. Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Eliminar:

- No hay flujo alternativo

Poscondiciones:

Crear:

- Se ha creado el comentario.

Actualizar:

- Se ha actualizado el comentario.

Eliminar:

➤ Se ha eliminado el comentario.

3.3.3 Actor estudiante.

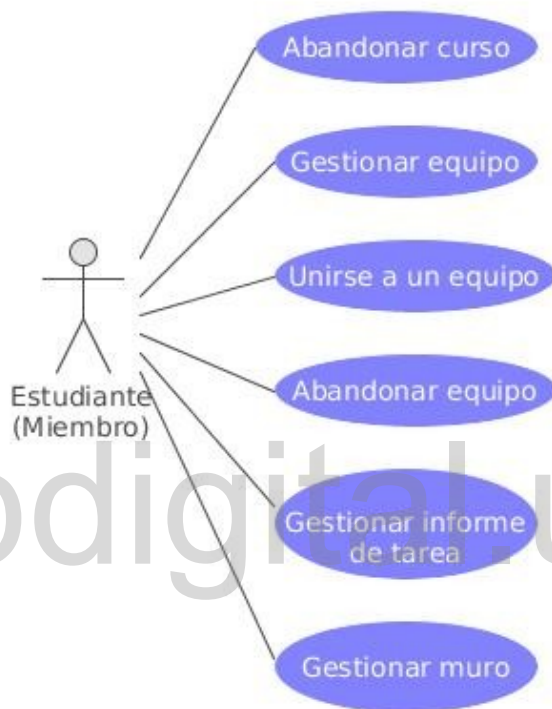


Figura 5: Casos de uso del estudiante

3.3.3.1 Abandonar curso

Nombre: Abandonar curso.
Descripción: Permite a un miembro de un curso desvincularse.
Actores: Estudiante miembro.
Precondiciones:

<ul style="list-style-type: none"> ➤ Debe haber iniciado sesión. ➤ Debe haber ingresado al curso donde se desempeña como miembro.
Flujo normal: <ul style="list-style-type: none"> ➤ 1. El estudiante presiona sobre el botón para desvincularse del curso. ➤ 2. El sistema valida y guarda los cambios
Flujo alternativo: <ul style="list-style-type: none"> ➤ 1.A El estudiante forma parte de un equipo por lo que no puede abandonar el curso.
Poscondiciones: <ul style="list-style-type: none"> ➤ 2. El estudiante ya no forma parte del equipo. ➤ 1A. El estudiante no puede abandonar curso

3.3.3.2 Gestionar equipo

Nombre: Gestionar equipo.
Descripción: Permite a un miembro de un curso crear un equipo.
Actores: Estudiante miembro.
Precondiciones: <ul style="list-style-type: none"> ➤ Debe haber iniciado sesión. ➤ Debe haber ingresado a un curso donde se desempeña como miembro. ➤ No debe pertenecer a ningún otro equipo dentro del curso.
Flujo normal: <u>Crear:</u> <ul style="list-style-type: none"> ➤ 1. El estudiante presiona sobre el enlace de crear equipo. ➤ 2. El sistema muestra el formulario que el estudiante debe llenar. ➤ 3. El estudiante ingresa los datos y pulsa el botón de guardar. ➤ El sistema valida, crea el equipo y el estudiante pasa ser líder dentro del curso.
Flujo alternativo:

➤ No hay flujo alternativo.
Poscondiciones: <u>Crear:</u> ➤ Se ha creado el equipo

3.3.3.3 Unirse a un equipo

Nombre: Unirse a un equipo.
Descripción: Permite a un miembro de un curso unirse a un equipo.
Actores: Estudiante miembro.
Precondiciones: <ul style="list-style-type: none"> ➤ Debe haber iniciado sesión. ➤ Debe haber ingresado a un curso donde se desempeña como miembro. ➤ No debe pertenecer a ningún otro equipo dentro del curso.
Flujo normal: <u>A través de una solicitud:</u> <ul style="list-style-type: none"> ➤ 1. El estudiante presiona sobre el botón para unirse al equipo. ➤ 2. El líder acepta la solicitud. ➤ 3. El sistema valida y registra al usuario en el curso.
Flujo alternativo: <u>A través de una solicitud:</u> <ul style="list-style-type: none"> ➤ 1.A El líder no acepta la solicitud.
Poscondiciones: <u>A través de una solicitud:</u> <ul style="list-style-type: none"> ➤ 3. El estudiante pasa a ser miembro del equipo. ➤ 1.A El estudiante aún no es miembro del equipo.

3.3.3.4 Abandonar equipo

Nombre: Abandonar equipo.
Descripción: Permite a un miembro de un equipo desvincularse.
Actores: Estudiante miembro.
Precondiciones: <ul style="list-style-type: none"> ➤ Debe haber iniciado sesión. ➤ Debe haber ingresado al equipo donde se desempeña como miembro.
Flujo normal: <ul style="list-style-type: none"> ➤ 1. El estudiante presiona sobre el botón para desvincularse del equipo. ➤ 2. El sistema valida y guarda los cambios
Flujo alternativo: <ul style="list-style-type: none"> ➤ 1.A El estudiante tiene tareas asignadas por lo que no puede abandonar el equipo.
Poscondiciones: <ul style="list-style-type: none"> ➤ 2. El estudiante ya no forma parte del equipo. ➤ 1A. El estudiante no puede abandonar equipo

3.3.3.5 Gestionar informe de tarea

Nombre: Gestionar informe de tarea.
Descripción: Permite al estudiante responsable de la tarea crear, editar y eliminar el informe relacionado con una tarea.
Actores: Estudiante miembro.
Precondiciones: <ul style="list-style-type: none"> ➤ Debe haber iniciado sesión.

- Debe haber ingresado al equipo donde se desempeña como miembro.
- Seleccionar la tarea donde es responsable.

Flujo normal:

Crear:

- C1. El sistema presenta un formulario que el estudiante debe llenar.
- C2. El estudiante edita y pulsa el botón guardar.
- C3. El sistema valida los datos y guarda los cambios.

Actualizar:

- A1. El estudiante pulsa el enlace para editar.
- A2. El sistema muestra un formulario.
- A3. El estudiante modifica y pulsa el botón guardar.
- A4. El sistema valida y guarda los cambios.

Eliminar:

- E1. El estudiante presiona el botón eliminar
- E2. El sistema chequea restricciones y elimina el informe.

Flujo alternativo:

Crear:

- C2..A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija

Actualizar:

- A3.A. Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija

Eliminar:

- No hay flujo alternativo

Poscondiciones:

Crear:

- Se ha creado el informe.

Actualizar:

- Se ha actualizado el informe.

Eliminar:

- Se ha eliminado el informe.

3.3.3.6 Gestionar muro

Nombre: Gestionar muro.
Descripción: Permite a los estudiantes interactuar con los demás miembros de un curso a través de la creación de pequeños comentarios.
Actores: Estudiante miembro.
Precondiciones: <ul style="list-style-type: none"> ➤ Debe haber iniciado sesión. ➤ Debe haber ingresado al curso donde se desempeña como miembro.
Flujo normal:
<u>Crear:</u> <ul style="list-style-type: none"> ➤ C1. El sistema presenta un formulario que el estudiante debe llenar. ➤ C2. El estudiante edita y pulsa el botón guardar. ➤ C3. El sistema valida los datos y guarda los cambios.
<u>Actualizar:</u> <ul style="list-style-type: none"> ➤ A1. El estudiante pulsa el enlace para editar, del comentario que creó. ➤ A2. El sistema muestra un formulario. ➤ A3. El estudiante modifica y pulsa el botón guardar. ➤ A4. El sistema valida y guarda los cambios.
<u>Eliminar:</u> <ul style="list-style-type: none"> ➤ E1. El estudiante presiona el botón eliminar, del comentario que creó. ➤ E2. El sistema chequea restricciones y elimina el comentario.
Flujo alternativo:
<u>Crear:</u>

- C2..A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Actualizar:

- A3.A. Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Eliminar:

- No hay flujo alternativo

Poscondiciones:

Crear:

- Se ha creado el comentario.

Actualizar:

- Se ha actualizado el comentario.

Eliminar:

- Se ha eliminado el comentario.

3.3.4 Actor estudiante líder.

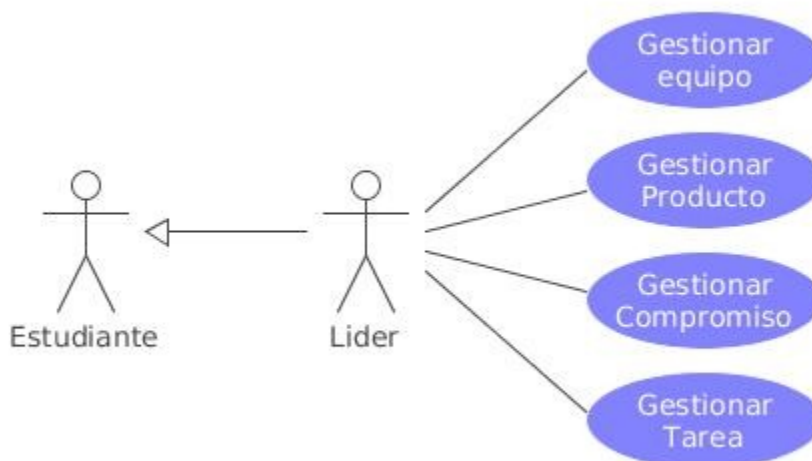


Figura 6: Casos de uso del estudiante líder

3.3.4.1 Gestionar equipo

Nombre: Gestionar equipo.
Descripción: Permite al líder del equipo actualizar y eliminar.
Actores: Estudiante líder.
Precondiciones: <ul style="list-style-type: none"> ➤ Debe haber iniciado sesión. ➤ Debe haber ingresado al equipo donde se desempeña líder.
Flujo normal: <u>Actualizar:</u> <ul style="list-style-type: none"> ➤ A1. El líder presiona sobre el enlace de editar equipo. ➤ A2. El sistema muestra el formulario que el estudiante debe llenar. ➤ A3. El estudiante ingresa los datos y pulsa el botón de guardar. ➤ A4. El sistema valida y guarda los cambios. <u>Eliminar:</u> <ul style="list-style-type: none"> ➤ E1. El líder presiona sobre el enlace de eliminar. ➤ E2. El sistema verifica las restricciones y elimina el equipo.
Flujo alternativo: <u>Actualizar:</u> <ul style="list-style-type: none"> ➤ A3.A. Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija. <u>Eliminar:</u> <ul style="list-style-type: none"> ➤ E1.A El equipo tiene miembros o productos, por lo que no se elimina.
Poscondiciones: <u>Actualizar:</u> <ul style="list-style-type: none"> ➤ Se ha actualizado el equipo. <u>Eliminar:</u> <ul style="list-style-type: none"> ➤ E2. Se elimina el equipo.

- E1.A No se puede eliminar el equipo

3.3.4.2 Gestionar producto

Nombre:
Gestionar producto.
Descripción:
Permite al líder del equipo crear, actualizar y eliminar un producto.
Actores:
Estudiante líder.
Precondiciones:
<ul style="list-style-type: none"> ➤ Debe haber iniciado sesión. ➤ Debe haber ingresado al equipo donde se desempeña líder.
Crear:
<ul style="list-style-type: none"> ➤ Puede heredar un producto prototipo, pero no editar o eliminar.
Flujo normal:
Crear:
<ul style="list-style-type: none"> ➤ C1. El líder presiona sobre el enlace para crear un producto. ➤ C2. El sistema muestra el formulario que el estudiante debe llenar. ➤ C3. El estudiante edita y presiona el botón de guardar. ➤ C4. El sistema valida y almacena los datos de entrada.
Actualizar:
<ul style="list-style-type: none"> ➤ A1. El líder accede al producto. ➤ A2. El líder presiona sobre el enlace de editar equipo. ➤ A3. El sistema muestra el formulario que el estudiante debe llenar. ➤ A4. El líder ingresa los datos y pulsa el botón de guardar. ➤ A5. El sistema valida y guarda los cambios.
Eliminar:
<ul style="list-style-type: none"> ➤ E1. El líder presiona sobre el enlace de eliminar. ➤ E2. El sistema verifica las restricciones y elimina el equipo.

Flujo alternativo:Crear:

- C4.A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Actualizar:

- A4.A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Eliminar:

- E1.A No se puede eliminar el producto porque tiene compromisos.

Poscondiciones:Crear:

- Se ha creado el producto.

Actualizar:

- Se ha actualizado el equipo.

Eliminar:

- E2. Se elimina el producto.
- E1.A No se puede eliminar el producto.

3.3.4.2 Gestionar compromiso**Nombre:**

Gestionar compromiso.

Descripción:

Permite al líder del equipo crear, actualizar y eliminar un compromiso relacionado con un producto.

Actores:

Estudiante líder.

Precondiciones:

- Debe haber iniciado sesión.
- Debe haber ingresado al equipo donde se desempeña líder.

- Si se trata de un producto prototipo los compromisos no pueden ser creados, editados o eliminados.

Actualizar:

- Solo puede actualizar la descripción pero no la fecha de entrega.

Flujo normal:

Crear:

- C1. El líder presiona sobre el enlace para crear un compromiso.
- C2. El sistema muestra el formulario que el estudiante debe llenar.
- C3. El estudiante edita y presiona el botón de guardar.
- C4. El sistema valida y almacena los datos de entrada.

Actualizar:

- A1. El líder accede al compromiso.
- A2. El líder presiona sobre el enlace de editar compromiso.
- A3. El sistema muestra el formulario que el estudiante debe llenar.
- A4. El líder ingresa los datos y pulsa el botón de guardar.
- A5. El sistema valida y guarda los cambios.

Eliminar:

- E1. El líder presiona sobre el botón de eliminar.
- E2. El sistema verifica las restricciones y elimina el compromiso.

Flujo alternativo:

Crear:

- C3.A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Actualizar:

- A4.A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Eliminar:

- E1.A No se puede eliminar el compromiso porque tiene tareas.

Poscondiciones:

Crear:

- Se ha creado el compromiso.

Actualizar:

- Se ha actualizado el compromiso.

Eliminar:

- E2. Se elimina el compromiso.
- E1.A No se puede eliminar el compromiso.

3.3.4.3 Gestionar tarea.

Nombre:

Gestionar tarea.

Descripción:

Permite al líder del equipo crear, actualizar y eliminar una tarea relacionada con un compromiso.

Actores:

Estudiante líder.

Precondiciones:

- Debe haber iniciado sesión.
- Debe haber ingresado al equipo donde se desempeña líder.
- Debe haber ingresado al compromiso.

Flujo normal:

Crear:

- C1. El líder presiona sobre el enlace para crear una tarea.
- C2. El sistema muestra el formulario que el estudiante debe llenar.
- C3. El estudiante edita y presiona el botón de guardar.
- C4. El sistema valida y almacena los datos de entrada.

Actualizar:

- A1. El líder accede a la tarea.
- A2. El líder presiona sobre el enlace de editar tarea.

- A3. El sistema muestra el formulario que el estudiante debe llenar.
- A4. El líder ingresa los datos y pulsa el botón de guardar.
- A5. El sistema valida y guarda los cambios.

Eliminar:

- E1. El líder presiona sobre el botón de eliminar.
- E2. El sistema verifica las restricciones y elimina la tarea.

Flujo alternativo:

Crear:

- C3.A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Actualizar:

- A4.A Los datos introducidos por el actor no pasan las validaciones, se le notifica y se presenta nuevamente el formulario para que corrija.

Eliminar:

- E1.A No se puede eliminar la tareas ya que se a editado un informe sobre esta.

Poscondiciones:

Crear:

- Se ha creado la tarea.

Actualizar:

- Se ha actualizado la tarea.

Eliminar:

- E2. Se elimina la tarea.
- E1.A No se puede eliminar la tarea.

3.4 Diagrama de actividades.

El proceso que más engloba actividades es el registro de compromisos prototipos.

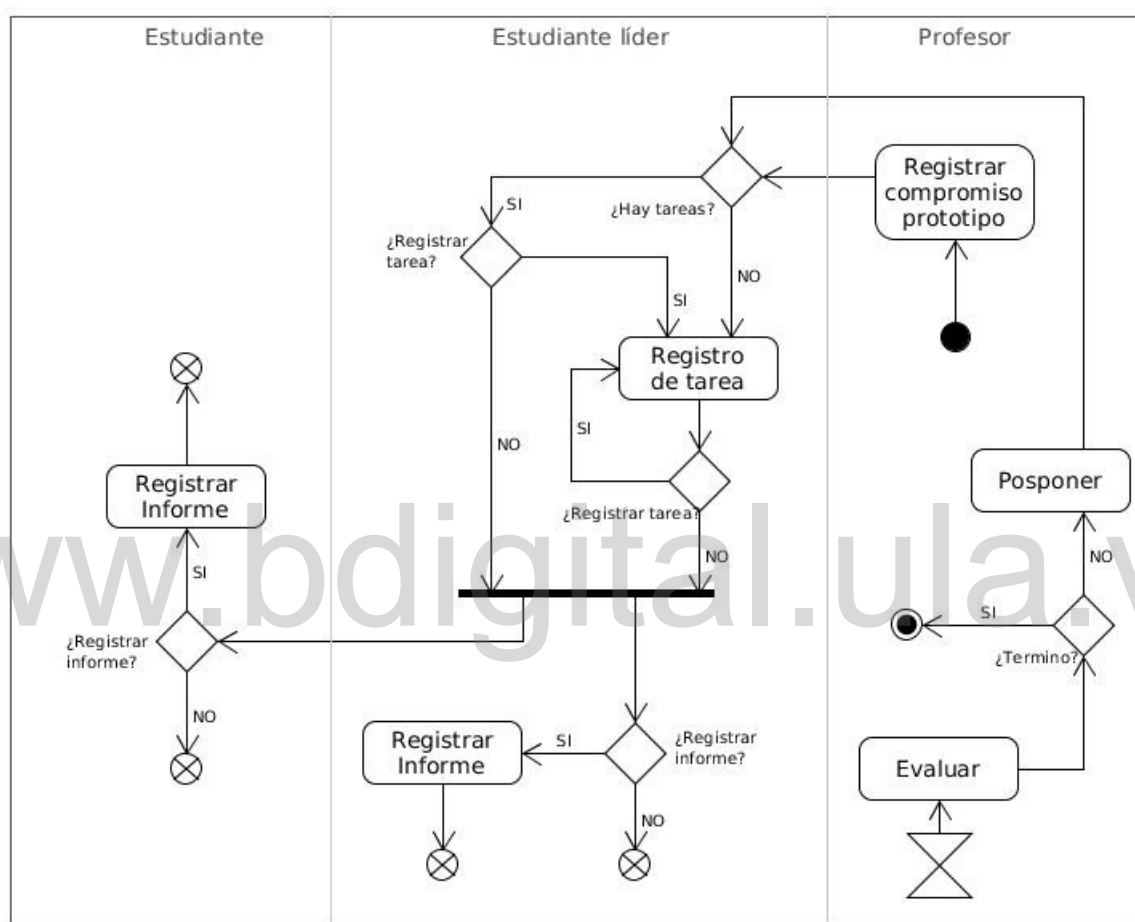


Figura 7: Diagrama de actividades para registro de compromisos

El jefe ejecutivo o profesor crea un compromiso prototipo a un producto prototipo lo que permite que el líder de los equipos que heredaron el producto prototipo asocien tareas sobre el compromiso prototipo y las asignen a algún miembro del equipo quien debe registrar un informe. El profesor evalúa el porcentaje de avance del compromiso y puede posponer el tiempo de entrega permitiendo editar el informe de la tarea.

Capítulo 4

Diseño y Arquitectura

4.1 Justificación y diseño.

Al tratarse de una nueva versión el sistema GeRAIS esta sigue los estándares establecidos en las versiones previas, ser desarrollado bajo una arquitectura web debido a lo practico que resulta el acceso a través de un navegador web, permitiendo además múltiples usuarios, con facilidad para actualizar, no depende del sistema operativo y sin la necesidad de distribuir e instalar software a potenciales usuarios. Para su uso sólo es necesario contar con un navegador capaz de ejecutar JavaScript.

La necesidad de avances continuos y prototipos funcionales definidos en la metodología llevada a cabo demanda contar con herramientas de desarrollo de software que proporcionen al programador mecanismos de rápida implementación para de esta forma reducir el tiempo de desarrollo.

Aunque las herramientas que fueron usadas en versiones anteriores de GeRAIS permiten ser actualizas a versiones superiores se opto por implementar nuevas herramientas que permitan desarrollar y obtener un producto mínimo viable de manera simple y rápidamente.

A continuación se exponen las herramientas utilizadas para el desarrollo de la nueva versión de GeRAIS:

- El lenguaje de programación utilizado es Ruby en su versión 2.4.0. Es un lenguaje de programación interpretado, reflexivo y orientado a objetos, además de dinámico y de código abierto enfocado en la simplicidad y productividad. Su alcance parece ilimitado y hoy se encuentra presente en aplicaciones que van desde el desarrollo web hasta la simulación de ambientes complejos. Gran parte de su crecimiento se atribuye

a la popularidad alcanzada por aplicaciones con el framework de desarrollo web Ruby on Rails.

- El framework utilizado es Ruby on Rails en su versión 5.0.1. Como se hizo alusión anteriormente está escrito en lenguaje de programación Ruby. Debe su éxito a su diseño elegante y compacto y crea de manera efectiva un lenguaje específico de dominio para escribir aplicaciones web, dando como resultado que tareas como generar HTML, crear modelos de datos y enrutar URL sean fáciles y un código conciso y legible.
- Uso de jQuery que viene integrado a Ruby on Rails. Esta es una librería, de código abierto, de JavaScript que permite interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar la técnica AJAX simplificando de esta manera la tarea de programar y agrega interactividad al sitio web.
- Uso Materialize, framework frontend, que está desarrollado en SASS y hace uso de las buenas prácticas en HTML5, CSS3 y JavaScript. Proporciona una forma fácil y rápida de implementar diseño web y funciona en la mayoría de los navegadores modernos. Está basado en el concepto de diseño de interfaces desarrollado e implementado por Google, llamado Material Design.
- El gestor de base de datos utilizado es MySQL en su versión 5.6.31. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.
- Firebase Cloud Messaging (FCM) es una solución de mensajería multiplataforma que permite enviar mensajes de forma segura y gratuita. Esta herramienta facilita el envío de notificaciones push.

4.2 Diagrama de despliegue.

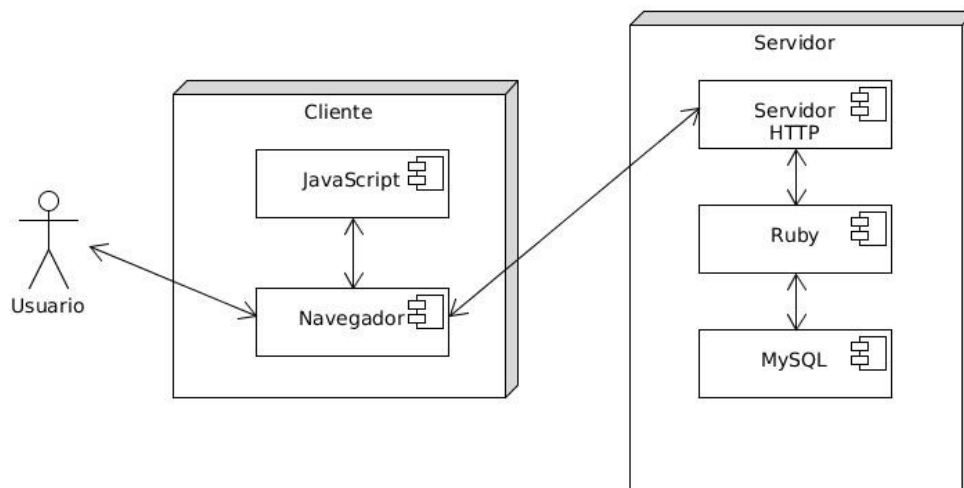


Figura 8: Diagrama de despliegue.

www.bdigital.ula.ve

4.3 Arquitectura de software.

Ruby on Rails sigue el patrón arquitectónico de modelo-vista-controlador (MVC), por tal razón este proyecto sigue dicho patrón, que impone una separación entre los datos de la aplicación (como la información del usuario) y el código para mostrarlo.

Al interactuar con una aplicación Ruby on Rails o simplemente Rails, un navegador envía una solicitud que es recibida por un servidor web y transmitida a un controlador de Rails, que se encarga de que hacer a continuación. En algunos casos el controlador renderizará inmediatamente una vista, que es una plantilla que se convierte en HTML y se devuelve al navegador. Más comúnmente para los sitios dinámicos, el controlador interactúa con el modelo, que es un objeto Ruby que representa un elemento del sitio (como usuarios) y se encarga de comunicarse con la base de datos. Después de invocar el modelo, el controlador entonces procesa la vista y devuelve la página web completa al navegador como HTML.

Las aplicaciones en Ruby on Rails tienen el siguiente flujo:

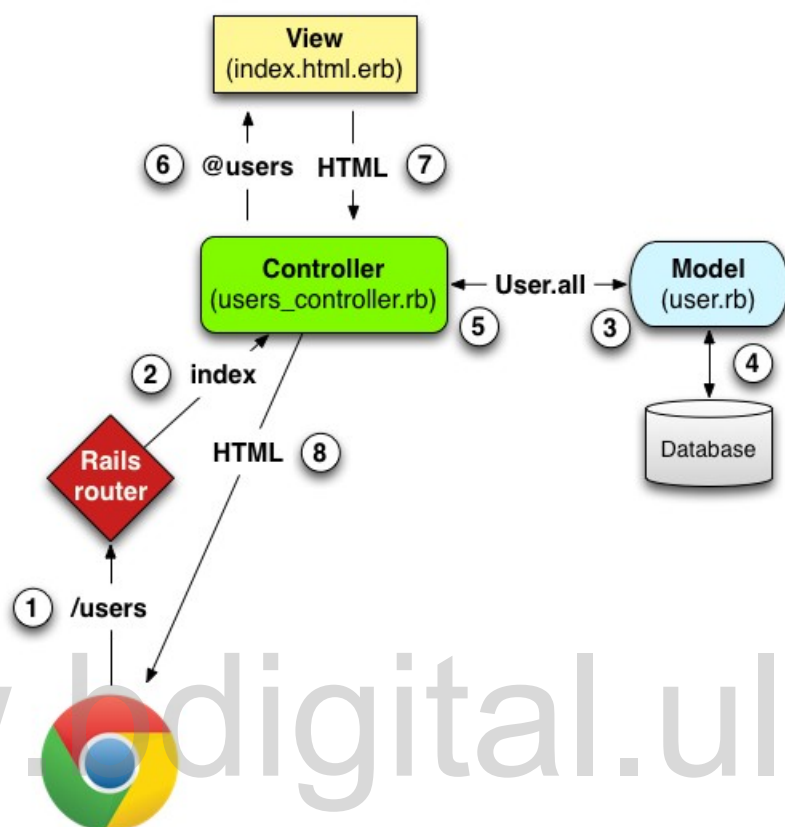


Figura 9: Flujo de una aplicación en Ruby on rails

1. El navegador emite una solicitud para la URL `/users`.
2. La ruta Rails `/users` para la acción `index` en el controlador de usuarios.
3. La acción `index` solicita al modelo de usuario que recupere a todos los usuarios (`User.all`).
4. El modelo de usuario extrae todos los usuarios de la base de datos.
5. El modelo de usuario devuelve la lista de usuarios al controlador.
6. El controlador captura los usuarios en la variable `@users` que se pasa a la vista `index`.
7. La vista utiliza Ruby incrustado para representar la página como HTML.
8. El controlador devuelve el código HTML al navegador.

4.4 Modelo relacional de la base de datos.

A continuación se muestra el esquema de la base de datos relacional:

- comments (id, comment_text, post_id, user_id, created_at, update_at).
- commitment_abstracts (id, abstract, commitment_id, created_at, update_at).
- commitment_prototypes (id, description, deadline, prototype_id, created_at, update_at).
- commitments (id, description, deadline, product_id, commitment_prototype_id, created_at, update_at).
- course_users (id, user_id, rol, course_id, team_id, created_at, update_at).
- courses (id, name, description, initials, period_type, section, category, institute, content, privacy, inscriptions_activated, evaluate_teacher, strict_mode_isa, code_confirmed, logo, period_length, code, created_at, update_at).
- fcm_token (id, token, user_id, created_at, update_at).
- notifications (id, noti_type, viewed, accepted, course_id, course_name, user_id, user_name, team_id, team_name, commitment_id, commitment_desc, task_id, task_desc, product_id, noti_user_id, created_at, update_at).
- posts (id, post_text, course_id, user_id, created_at, update_at).
- product_reports (id, report, product_id, created_at, update_at).
- products (id, name, description, team_id, logo, initials, prototype_id, created_at, update_at).
- prototypes (id, name, description, course_id, logo, initials).
- tasks (id, description, execution, commitment_id, user_id, due_date, created_at, update_at).
- tasks_abstracts (id, abstract, task_id, created_at, update_at).
- teams (id, name, description, initials, logo, course_id, created_at, update_at).
- users (id, names, lastnames, email, password_digest, initials, country, city, phone, sn_one, sn_two, skills, reset_digest, reset_sent_at, image_user, created_at, update_at).

Analizando las dependencias funcionales se puede determinar lo siguiente:

1. No existen en el modelo relacional atributos multi-evaluados y todas las tuplas tienen el mismo número de columnas, por lo tanto, está en primera forma normal.
2. Los atributos de todas las tablas dependen totalmente de la clave primaria y está en primera forma normal, por lo tanto, está en segunda forma normal.
3. No existen dependencias transitivas en el modelo y está en segunda forma normal, por lo tanto, está en tercera forma normal.

www.bdigital.ula.ve

4.5 Diagrama de clases.

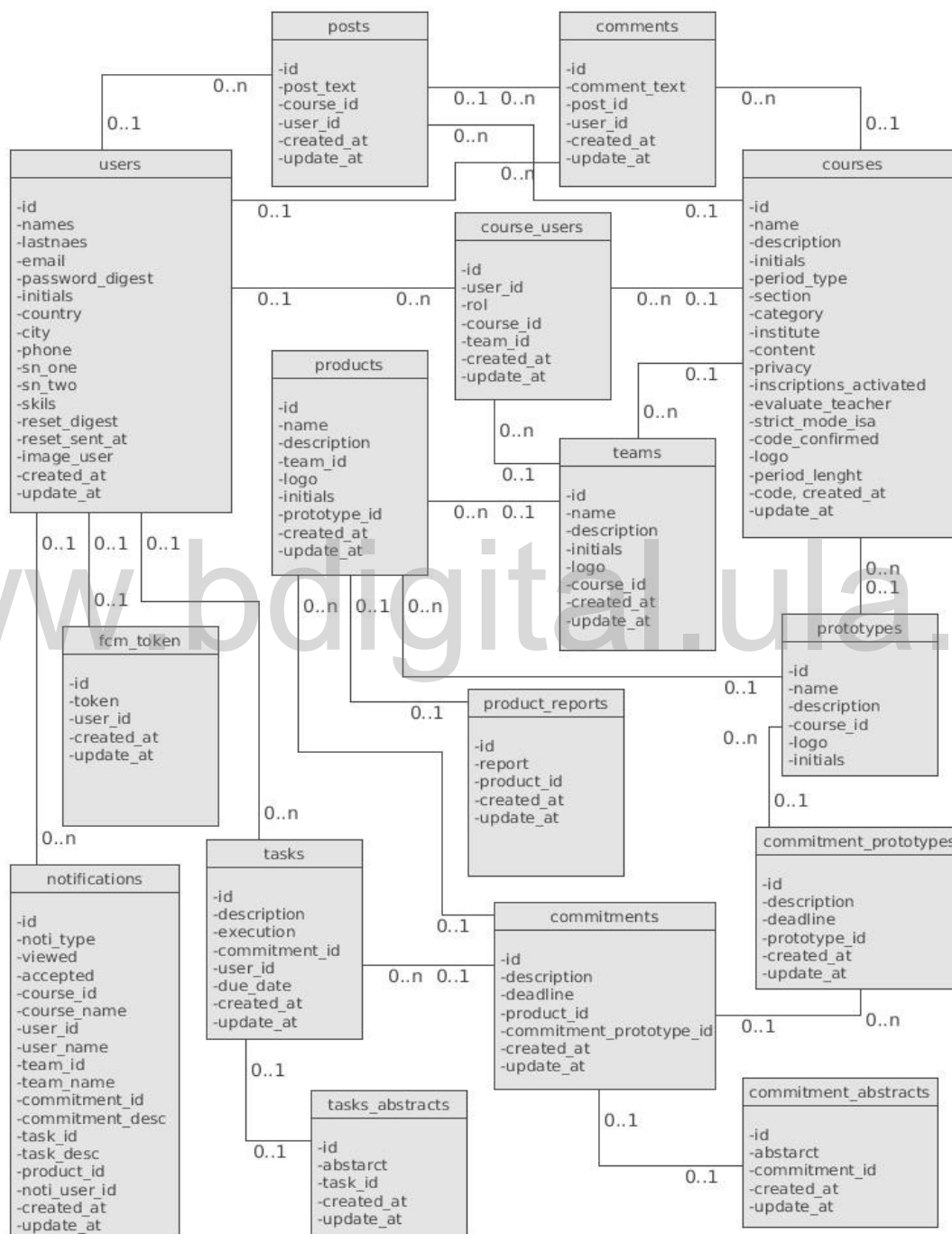


Figura 10: Diagrama de clases

4.6 Diagrama entidad relación extendido de la base de datos.

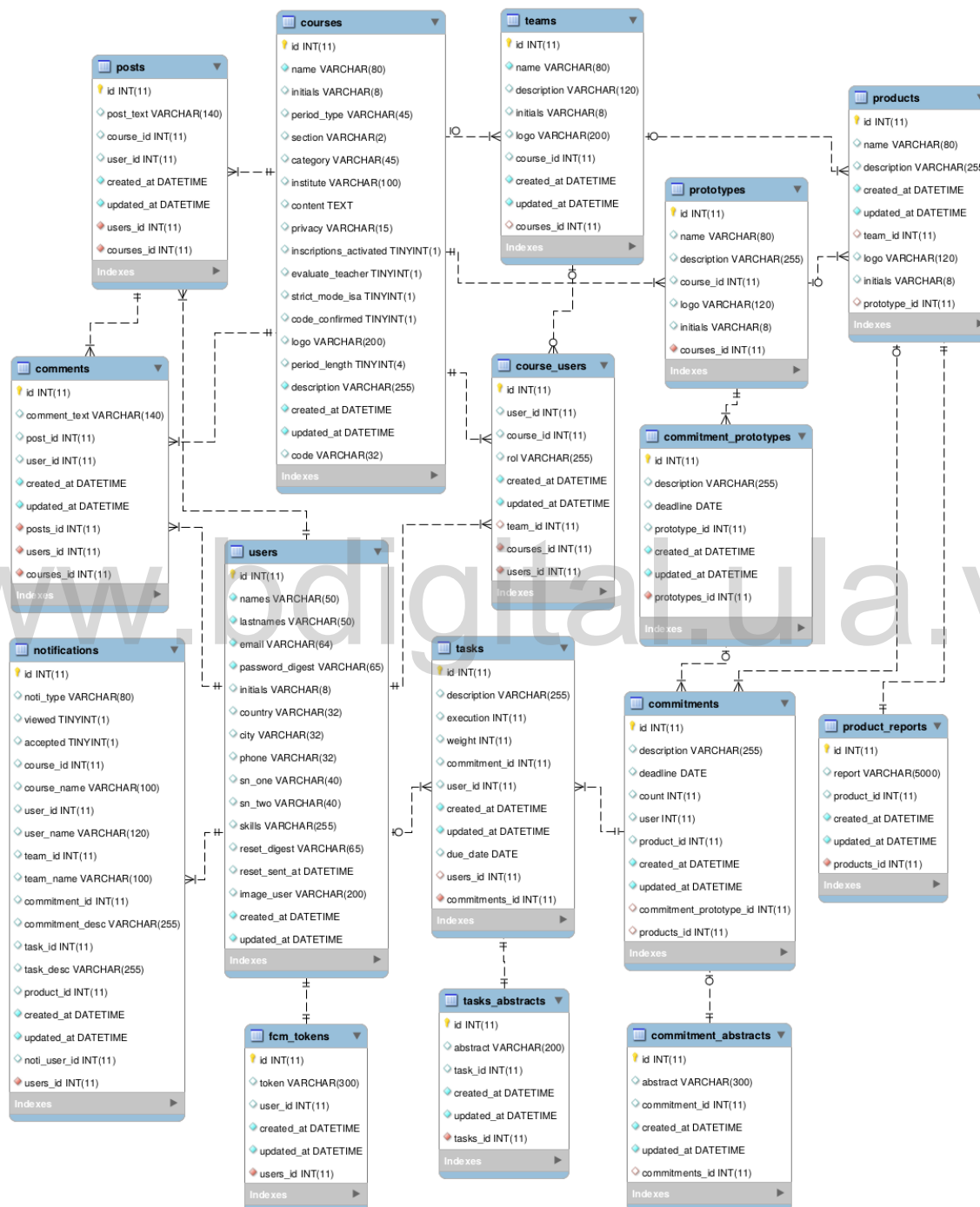


Figura 11: Diagrama ERE de la base de datos

4.7 Estructura de la base de datos.

A continuación se muestra la descripción de cada una de las tablas propuestas en GeRAIS versión 3.0.

4.7.1 Tabla de comentarios (comments)

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
comment_text	varchar(140)	Comentario
post_id	int(11) unsigned	Apuntador al post
user_id	int(11) unsigned	Apuntador al usuario
created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 1: Tabla de comentarios

4.7.2 Tabla de informe de compromiso (commitment_abstracts)

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
abstract	varchar(300)	Informe del compromiso
commitment_id	int(11) unsigned	Apuntador a compromiso
created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 2: Tabla de informe de compromiso

4.7.3 Tabla de compromiso prototipo (commitment_prototypes)

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
description	varchar(255)	Descripción

deadline	date	Fecha limite
prototype_id	int(11) unsigned	Apuntador a prototipo
created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 3: Tabla de compromisos prototipos

4.7.4 Tabla de compromisos (commitments)

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
description	varchar(255)	Descripción
deadline	date	Fecha limite
prototype_id	int(11) unsigned	Apuntador a producto
commitment_prototype_id	int(11) unsigned	Apuntador a compromiso prototipo
created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 4: Tabla de compromisos

4.7.5 Tabla de curso-usuarios (course_users)

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
rol	varchar(255)	Rol del usuario
course_id	int(11) unsigned	Apuntador a curso
team_id	int(11) unsigned	Apuntador a equipo
created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 5: Tabla curso-usuarios

4.7.6 Tabla de cursos (courses)

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
name	varchar(80)	Nombre
description	varchar(255)	Descripción del curso
initials	varchar(8)	Iniciales del nombre
period_type	varchar(45)	Periodo
section	varchar(2)	Sección
category	varchar(45)	Categoría del curso
institute	varchar(45)	Nombre de una institución
content	text	Contenido del curso
privacy	varchar(15)	Tipo de curso, privado o público.
inscriptions_activated	tinyint(1)	Inscripciones activas o desactivadas.
evaluate_teacher	tinyint(1)	Se evalúa o no al profesor
strict_mode_isa	tinyint(1)	Informes estrictos
code_confirmed	tinyint(1)	Código para inscripción
logo	varchar(200)	Extensión de la imagen
period_length	tinyint(4)	Duración del curso-usuario
code	varchar(32)	Código del curso
created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 6: Tabla de cursos

4.7.7 Tabla de identificador de Firebase (fcm_token)

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
token	varchar(300)	Identificador

user_id	int(11)	Apuntador a usuario
created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 7: Tabla identificador firebase

4.7.8 Tabla de notificaciones (notifications)

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
noti_type	varchar(80)	Tipo de notificación
viewed	tinyint(1)	Notificación vista
accepted	tinyint(1)	Notificación aceptada
course_id	int(11) unsigned	Apuntador a curso
course_name	varchar(100)	Nombre del curso
user_id	int(11) unsigned	Apuntador a usuario
user_name	varchar(120)	Nombre del usuario
team_id	int(11) unsigned	Apuntador a equipo
team_name	varchar(100)	Nombre del equipo
commitment_id	int(11) unsigned	Apuntador a compromiso
commitment_desc	varchar(255)	Descripción del compromiso
task_id	int(11) unsigned	Apuntador a tarea
task_desc	varchar(255)	Descripción de la tarea
product_id	int(11) unsigned	Apuntador a producto
noti_user_id	int(11) unsigned	Id del usuario
created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 8: Tabla de notificaciones

4.7.9 Tabla de posts

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
post_text	varchar(140)	Comentario
course_id	int(11) unsigned	Apuntador de curso
user_id	int(11) unsigned	Apuntador de usuario
created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 9: Tabla de posts

4.7.10 Tabla de informe de productos (product_reports)

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
report	varchar(5000)	Descripción del producto
product_id	int(11) unsigned	Apuntador a producto
created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 10: Tabla de informe de productos

4.7.11 Tabla de productos(products)

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
name	varchar(80)	Nombre del producto
description	varchar(255)	Descripción del producto
team_id	int(11) unsigned	Apuntador al equipo
prototype	int(11) unsigned	Apuntador a prototipo
logo	varchar(120)	Extensión de la imagen
initials	varchar(8)	Iniciales del nombre

created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 11: Tabla de productos

4.7.8 Tabla de productos prototipos (prototypes)

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
name	varchar(80)	Nombre del producto prototipo
description	varchar(255)	Descripción del producto prototipo
course_id	int(11) unsigned	Apuntador a curso
logo	varchar(120)	Extensión de la imagen
initials	varchar(8)	Iniciales del nombre

Tabla 12: Tabla de productos prototipos

4.7.8 Tabla de tareas (tasks)

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
description	varchar(255)	Descripción de la tarea
execution	int(11) unsigned	Porcentaje de avance de la tarea
commitment_id	int(11) unsigned	Apuntador al compromiso
user_id	int(11) unsigned	Apuntador al usuario
due_date		Fecha de límite
created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 13: Tabla de tareas

4.7.8 Tabla de informe de tareas

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
abstract	varchar(200)	Descripción del informe
task_id	int(11) unsigned	Apuntador a la tarea
created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 14: Tabla de informe de la tarea

4.7.8 Tabla de equipos (teams)

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
name	varchar(80)	Nombre del equipo
description	varchar(120)	Descripción del equipo
logo	varchar(200)	Extensión de la imagen
initials	varchar(8)	Iniciales del nombre
course_id	int(11) unsigned	Apuntador al curso
created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 15: Tabla de equipos

4.7.8 Tabla de usuarios (users)

Nombre	Tipo	Descripción
id	int(11) unsigned	Clave primaria
names	varchar(50)	Nombres del usuario
lastnames	varchar(50)	Apellidos del usuario
email	varchar(64)	Correo electrónico
password_digest	varchar(65)	Contraseña

initials	varchar(8)	Iniciales del nombre
country	varchar(32)	País
city	varchar(32)	Ciudad
phone	varchar(32)	Número de teléfono
sn_one	varchar(40)	Red social
sn_two	varchar(40)	Red social
skills	varchar(255)	Talentos
reset_digest	varchar(65)	Contraseña de reactivación
reset_sent_at	datetime	Fecha para contraseña
image_user	varchar(200)	Extensión de la imagen
created_at	datetime	Fecha de creación
update_at	datetime	Fecha de actualización

Tabla 16: Tabla de usuarios

4.8 Diseño de la interfaz de usuario.

Para el diseño de GeRAIS se implementó el framework front-end Materialize que esta basado en el concepto Material Design, que es un diseño de interfaces desarrollado e implementado por Google, donde la profundidad, las superficies, los bordes, las sombras, colores y movimientos juegan un papel principal. También se hace uso de iconos de Google, a través de la gema material_icons, los cuales también están enmarcados en el concepto Material Design.

Materialize provee de varios componentes predefinidos que van desde pantallas modales, botones, formularios, menú, preloaders, tablas y muchos más. Igualmente cuenta con un sistema de rejillas (grid) responsive adaptable a pantallas y como está desarrollado en SASS muchos de estos componentes pueden ser editables. También fueron diseñados algunos componentes, haciendo uso de SASS, para mostrar una interfaz lo más similar a la desarrollada en la versión móvil Android, en la tesis: Desarrollo de una aplicación móvil Android para la gestión de productos RAIS.

A continuación se presenta capturas de pantalla donde se muestra los elementos que componen las interfaces:

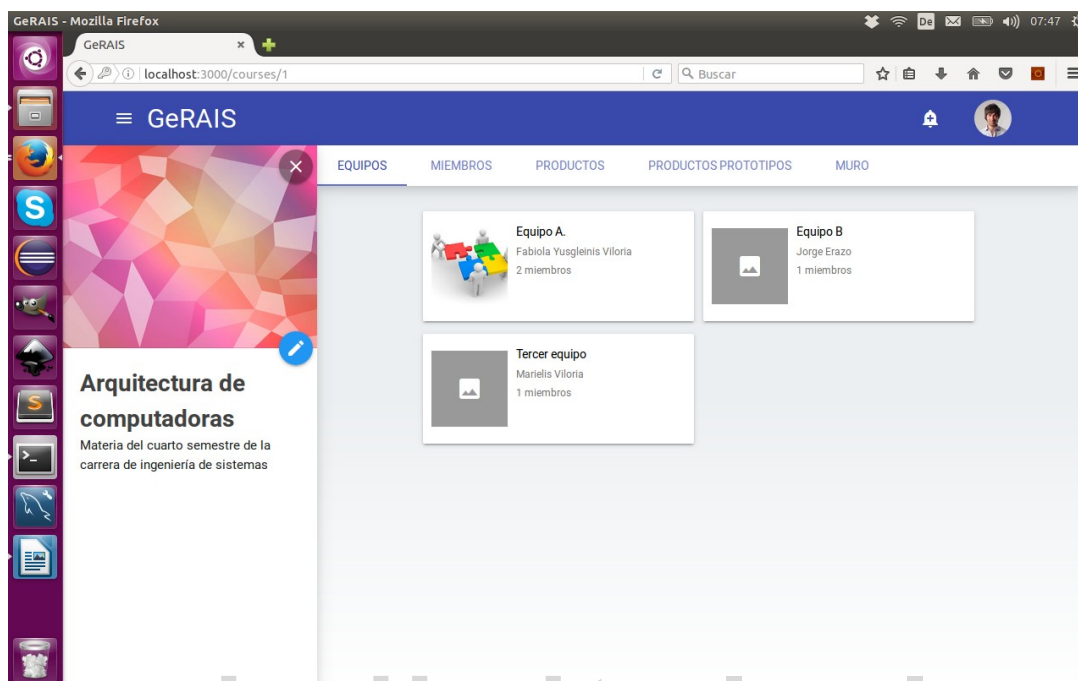


Figura 12: Vista de un curso



Figura 13: Vista de curso para movil-web

En las figuras anteriores se muestran los elementos más comunes que se pueden encontrar en la interfaz de usuario.

Encabezado: Ubicado en la parte superior de la página, muestra al usuario la identificación de la herramienta y enlaces directos a notificaciones, editar perfil y cerrar sesión.

Sub-encabezado: Ubicado en la parte superior seguidamente del encabezado, muestra las secciones que componen el ambiente de trabajo.

Barra lateral: Muestra información referente al ambiente de trabajo, no disponible en la versión web móvil.

Cuerpo de la página: se presenta información referente a las secciones que componen el ambiente de trabajo.

www.bdigital.ula.ve

Capítulo 5

Especificaciones, desarrollo e implementación.

Desarrollo de GeRAIS versión 3.0.

Antes que nada, para el desarrollo de versión 3.0 de la herramienta GeRAIS se accedió a la versión 2.0 con el fin de estudiar su comportamiento y funcionalidades. El código de dicha versión está alojado en el sistema de control de versiones proporcionado por GitHub en el siguiente enlace: <https://github.com/FernandoOsuna/gerais-v2>

En la versión 3.0 no se mantendrá ninguno de los frameworks empleados en versiones anteriores. En esta ocasión se opta por el uso del framework Ruby on Rails escrito en el lenguaje de programación Ruby.

El framework Ruby on Rails sigue el paradigma del patrón Modelo Vista Controlador (MVC) y las piezas de dicha arquitectura son las siguientes:

- El modelo: que consiste en clases que representan a las tablas de la base de datos. En Ruby on Rails son gestionadas por ActiveRecord, que incluye adaptadores de base de datos para conectarse a SQLite, MySQL y PostgreSQL. Por lo general, se hereda una de las clase ApplicationRecord, esta a su vez lo hace de ActiveRecord::Base, y el programa averigua automáticamente que tabla usar y las columnas que posee.

Las definiciones de las clases también detallan las relaciones entre clases con sentencias de mapeo de objeto relacional. Por ejemplo, si la clase *Image* tiene una definición `has_many:comentarios`, y existe una instancia de *Image* llamada *img*, entonces `img.comentarios` devolverá un array con todos los objetos *Comentario* cuya columna *image_id* (en la tabla comentarios) sea igual a *img.id*.

Las rutinas de validación de datos (validates) y las rutinas relacionadas con la actualización (after_destroy, before_update), que ejecutan métodos antes de una acción determinada, también se especifican e implementan en la clase del modelo.

Los modelos están ubicados en el directorio /app/models y en GeRAIS se distinguen los siguientes:

- comment
 - commitment
 - commitment_abstract
 - commitment_prototype
 - course
 - course_user
 - notification
 - post
 - product
 - product_report
 - product_user
 - prototype
 - task
 - tasks_abstract
 - team
 - user
-
- Las vistas: es como se muestran los datos de las clases de controlador, generalmente consiste en una cantidad de código incluido en HTML. El método empleado por Rails por defecto es Ruby Empotrado que son fragmentos de código HTML con algo de código en Ruby. Las vistas se encuentran ubicadas en el directorio /app/views y están relacionadas con las acciones o métodos de los controladores que ameriten mostrar información. Entre las vistas más comunes de la herramienta GeRAIS se distinguen: index y show.

- El controlador: este responde a la interacción del usuario e invoca a la lógica de la aplicación, que a su vez manipula los datos de las clases del Modelo y muestra los resultados usando las vistas. El controlador hereda de la clase ApplicationController y define las acciones necesarias a través de métodos, que pueden ser invocados desde la web por lo general `http://aplication/ejemplo/metodo`, que invoca a la clase controlador Ejemplo y al método definido en esta y presenta los datos usando la plantilla `/app/views/ejemplo/método.html.erb`.

En esta versión se hace uso de un conjunto de métodos o acciones desarrollados para la versión móvil android, de la tesis: Desarrollo de una aplicación móvil Android para la gestión de productos RAIS, las cuales sirven a peticiones haciendo uso de el formato JSON, con el fin de no redundar en el código.

El directorio de los controladores es `/app/controllers` y en GeRAIS se han definido los siguientes:

- `comments_controller`
- `commitment_abstracts_controller`
- `commitments_controller`
- `commitments_prototype_controller`
- `courses_controller`
- `dashboard_controller`
- `fcm_tokens_controller`
- `notifications_controller`
- `password_resets_controller`
- `posts_controller`
- `products_controller`
- `prototypes_controller`
- `sessions_controller`
- `tasks_abstracts_controller`
- `tasks_controller`
- `teams_controller`

- `users_controller`

Además de los componentes anteriormente mencionados también se tiene una serie de ayudantes o helpers, ubicados en `/app/helpers` los cuales son funciones que pueden ser utilizados en cualquiera de las capas y en cualquier momento.

Ruby on Rails también permite incorporar gemas, que son programas o bibliotecas en Ruby, que enriquecen o facilitan el desempeño de la herramienta. Todas las gemas utilizadas en la herramienta GeRAIS se encuentran en el archivo `Gemfile` en la carpeta `grais`. Entre las que se pueden contar están `jquery-rails` para la incorporación de JQuery, `mysql2` para facilitar el manejo de la base de datos a través de Mysql, `material_icons` para la incorporación de los iconos de Google, `will_paginate` para facilitar la tarea de la paginación entre las vistas, entre otras.

De esta manera, cada solicitud realizada por el usuario a través de una ruta decide qué acción de qué controlador recibe qué petición. El método o acción del controlador realiza la consulta a la base de datos a través del modelo que está asociado una tabla de la misma y determina en qué formato devolver la información que será desplegada en la vista que por lo general lleva el mismo nombre de la acción. La vista implementa el framework front end Materialize que permite mostrar la información con un estilo más agradable al usuario, mientras que el uso de JQuery, herramienta JavaScript que proporciona llamadas Ajax y otras funcionalidades, aporta el dinamismo necesario a la hora de interactuar pues permite enviar o obtener información a través del formato JSON o JavaScript sin la necesidad de recargar la página como normalmente funciona con el formato HTML. El uso de JavaScript también permite la incorporación de Firebase Cloud Messaging de esta forma recibir mensajes de notificación en las apps web que se ejecutan en navegadores que admiten service worker.

Capítulo 6

Conclusiones y Recomendaciones.

6.1 Conclusiones.

Después de indagar en qué consistía la estrategia RAIS, componentes, características y como se implementa en el salón de clase, se procedió a evaluar la versión 2.0 de la herramienta tecnológica GeRAIS, basada en dicha estrategia, así como otras herramientas tecnológicas que son utilizadas actualmente. Esto permitió identificar las deficiencias y restricciones que poseía la herramienta GeRAIS y dar paso a definir los requerimientos tecnológicos para desarrollar un sistema de gestión que fuese óptimo y robusto.

Es así como se diseñó y desarrolló una nueva versión para la herramienta GeRAIS bajo el patrón de arquitectura web cliente-servidor, que brinda la posibilidad de integrar múltiples usuarios y múltiples cursos, sin la necesidad de un administrador, poder recuperar la contraseña del usuario, así como un canal de comunicación de fácil acceso y uso, debido a la adecuada disposición de los elementos en la interfaz, alcanzándose de esta manera el objetivo general del trabajo.

Esta nueva versión de la herramienta GeRAIS presenta características, funcionalidades y una interfaz de usuario que modifica la forma de interactuar con esta. Se cuentan con enlaces directos a los diferentes ambientes de trabajo de la herramienta que permiten al usuario acceder y trabajar de forma directa. Cuenta con elementos como barrar laterales, tarjetas (cards) o etiquetas (tabs) que proporcionan al usuario la información necesaria para identificar el ambiente de trabajo en el cual se desenvuelve. Para la creación, modificación o eliminación de los elementos se muestran enlaces o botones que posibilitan identificarlos fácilmente.

A continuación se describen los cambios más significativos:

- La posibilidad de cualquier usuario de crear una cuenta sin la necesidad de pasar antes por la aprobación de un administrador.
- Un usuario registrado puede crear uno o varios cursos, en los cuales se desempeñaría como profesor, o formar parte de uno o varios cursos, en los cuales se desempeñaría como estudiante.
- Enlaces directos a los componentes o ambientes de trabajo de la herramienta relacionados con el usuario en sesión.
- Enviar solicitudes de ingreso a un curso o equipo haciendo uso del sistema de notificaciones push de Google.
- Ingresar a un curso a través de un código de validación que la herramienta genera a la hora de crear el curso.
- Recuperar la contraseña de usuario a través del envío de una notificación al correo electrónico.
- Se establece la figura de equipo en lugar de compañía.
- Se cuenta con un canal de comunicación por cada curso, donde solo los miembros del curso pueden crear, editar y eliminar comentarios.
- El líder del equipo tiene la posibilidad de crear un producto o heredar de uno propuesto por el profesor.
- El estudiante líder propondrá los compromisos y su fecha de entrega, pero solo profesor podrá editar la fecha de entrega de los compromisos.
- Solo el profesor podrá editar el porcentaje de avance de una tarea.
- El líder es el encargado de crear las tareas y asignarlas a los miembros del equipo.
- No solo las tareas contarán con un informe, también existe la posibilidad de crear uno para cada compromiso y uno para el producto, bajo la responsabilidad del líder.
-

Tener a disposición una serie de aplicaciones o herramientas de software como foros, correos electrónicos, grupos de redes sociales, entre otros, conduce a que muchos de quienes hace uso de la estrategia RAIS las implementen para hacer el seguimiento de los proyectos.

Al contar con una variedad tan amplia de herramientas es posible que muchos desistan de aplicar la estrategia al no contar con una que este enfocada en las especificaciones propias de RAIS.

De manera tal se puede deducir que GeRAIS en su versión 3.0 sirve como estímulo para que estrategia RAIS sea utilizada pues da soporte a muchas de las necesidades de dicha estrategia, facilitando su implementación.

6.2 Recomendaciones.

A pesar de que el sistema GeRAIS es capaz de responder a las peticiones de la versión web y la versión móvil no es la forma óptima de hacerlo, pues se está redundando en el código con el fin de acoplarse a las necesidades de cada versión, pudiendo delegar mucha de esta lógica a los modelos. Además, como la herramienta tiene potencial de escalabilidad, llegará un punto en que las interfaces de la versión web y móvil no coincidan del todo y por ende demande consumir datos de otra manera; así que se presenta como una alternativa crear una API REST que es un conjunto de rutinas que provee acceso a funciones de un determinado software usando como lenguaje de intercambio XML o JSON y contar con aplicación web, la cual es diseñada e implementada para ser consumida y usada por usuarios finales, donde los datos pretenden que se vean directamente produciendo un documento HTML (con CSS y / o Javascript).

Es importante señalar que la integración de la biblioteca de JavaScript para las notificaciones push de Google, no funciona adecuadamente, generando problemas declaración de variables a medida que se navega por la herramienta, aunque se procedió como lo indica las instrucciones. Sería de gran utilidad mejorar este aspecto y contar con un sistema de notificaciones propio de la herramienta.

En aras de una mayor escalabilidad sería ideal contar con una herramienta que no solo este en Español, que aunque facilite el acceso y apropiación por parte de los hispanohablantes, también obstaculiza el potencial de crecimiento que puede brindar tener una herramienta traducida a uno o varios idiomas adicionales, como por ejemplo inglés.

De igual manera a nivel de administración o gestión de tareas sería de gran utilidad poder reasignar una tarea a otro miembro del equipo, en caso de que el responsable no

cumpla o abandone el equipo. Así como también poder adjuntar archivos para respaldar el trabajo hecho.

En la gestión de cursos permitir que el jefe ejecutivo tenga la posibilidad de retirar a miembros del curso, también aplicaría la posibilidad para el líder de cada equipo de retirar cuando considere necesario a integrantes del mismo.

www.bdigital.ula.ve

Bibliografía.

Sandia, B., Gutiérrez, D., Hernández, D., Páez Monzón, G., Alvarado, J., Parada Puig, J. E., & Vergara Paredes, M. J. (20 de Septiembre de 2010). RAIS: Una Estrategia para el Manejo Integral de Conocimiento. Enseñanza de la Ingeniería Reproduciendo un Ambiente Industrial. Recuperado el 10 de Enero de 2017, de <http://saber.ula.ve/handle/123456789/33758>

Pressman, R. (2005). Ingeniería del Software. Un enfoque práctico. España: Mc Graw Hill.

Osuna, A. (2015). GeRAIS.2: Desarrollo de la versión 2.0 del Sistema para el Manejo y Seguimiento de productos RAIS (Reproducción de un Ambiente de Innovación en el Salón de clases). Universidad de Los Andes, Mérida.

Hartl, M. (2016). Ruby on Rails Tutorial, Learn Web Development with Rails [Libro en línea]. Consultado el 22 de septiembre de 2017 en: www.railstutorial.org/book

Ruby on Rails Guides (2017). [Pagina web en línea]. Disponible en: <http://guides.rubyonrails.org/index.html>

Cwesley, A(2013). UML Destilled (3ª de).

Elmasri, R., & Navathe, S. (2007). Fundamentos de Sistemas de Bases de Datos. Madrid: Pearson (2ª ed)

Rumbaugh, J; Jacobson, I y Booch, G. (2000). El lenguaje unificado de modelado. Manual de referencia.(1ª de)

Korth, H. F., & Silberschatz, A. (2002). Fundamentos de bases de datos. Madrid (4ª ed)

Flanagan, D.(2011). JavaScript: The Definitive Guide (6ª ed)

M, Seyed; “Saied” Tahaghoghi y Williams H.(2007). Learning MySQL (1ª de)

Powers, D.(2012). Beginning CSS3.

Andrew, R. (2009). The CSS Anthology: 101 Essential Tips, Tricks & Hacks (3ª ed)

Gauchat , J (2012). El gran libro de HTML5, CSS3 y Javascript (1ª ed)

Colaboradores de Wikipedia. (5 Enero de 2017). Usabilidad. Recuperado el 10 de Enero de 2017, de <https://es.wikipedia.org/wiki/Usabilidad>

Colaboradores de Wikipedia. (2 Enero de 2017). Aplicación web. Recuperado el 10 de Enero de 2017, de https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web

The jQuery Foundation. (2017). JQuery. Recuperado el 22 de Septiembre de 2017, de <http://jquery.com/>

Cómo configurar una app cliente de Firebase Cloud Messaging en JavaScript. (2017). [Página web en línea]. Disponible en: <https://firebase.google.com/docs/cloud-messaging/js/client?hl=es-419>

Materialize. (2017). [Pagina web en línea]. Disponible en: <http://materializecss.com>

Colaboradores de Wikipedia. (03 de Abril de 2017). Historias de usuario. Recuperado el 15 de Agosto de 2017, de https://es.wikipedia.org/wiki/Historias_de_usuario

Learn HTML, Learn CSS, Learn JavaScript. (2017). [Pagina web en línea]. Disponible en: www.w3schools.com

www.bdigital.ula.ve

Apéndice 1

Glosario y definiciones.

8.1 Definiciones técnicas.

A continuación se explican los conceptos necesarios para entender los detalles del desarrollo del sistema.

8.1.1 HTML

En relación al significado de HTML, Robbins (2013) señala:

HTML (HyperText Markup Language) es el lenguaje de marcado utilizado para convertir documentos de texto en páginas web y aplicaciones. El propósito fundamental de HTML como lenguaje de marcado es proporcionar una descripción semántica (el significado) del contenido y establecer una estructura de documento (una jerarquía de elementos).

8.1.2 Navegador web

Santos García & Sánchez García (citado por Osuma 2015) señalan que son aplicaciones de escritorio capaces de interpretar y presentar contenidos en hipertexto al usuario permitiéndole interactuar con él. El nombre de navegador hace referencia a que los contenidos en hipertexto normalmente están cargados con hipervínculos que permiten enlazar otros contenidos que a su vez también tienen otros hipervínculos. Al pasear por los mismos se podría decir que se está navegando en el mar de la información; esa es la razón de su nombre. Existen muchas aplicaciones de este tipo, disponibles para casi todas las plataformas disponibles, entre las cuales tenemos Firefox, Safari, Opera, Internet Explorer entre muchos otros.

8.1.3 Framework

Según Basandra (citado por Osuma, 2015) se trata de una estructura o conjunto de reglas que se usan como base para abordar o resolver problemas complejos.

8.1.4 UML

“El lenguaje de modelado unificado (UML) es una familia de anotaciones gráficas, respaldadas por un meta-modelo único, que ayudan a describir y diseñar sistemas de software, en particular los sistemas de software construidos con el estilo orientado a objetos (OO).”(Fowler, 2003, p. 14).

8.1.5 MySQL

Oracle Corporation (citado por Osuma, 2015) define a MySQL como un sistema de gestión de base de datos multiusuario propiedad de Oracle, se distribuye bajo doble licencia siendo una de ellas libre, utiliza SQL para el manejo de los datos y posee interfaces para conectarse desde múltiples lenguajes.

8.1.6 Ruby

Un lenguaje de programación dinámico y de código abierto enfocado en la simplicidad y productividad. Su elegante sintaxis se siente natural al leerla y fácil al escribirla. En Ruby todo es un objeto. A diferencia de otros lenguajes de programación orientada a objetos, Ruby se caracteriza por su intencional herencia simple. Sin embargo, Ruby incorpora el concepto de módulos, que son colecciones de métodos.(Comunidad Ruby, 2017).

8.1.7 Ruby on Rails

Con respecto a Ruby on Rails, Kehoe (2016), expresa lo siguiente:

Rails es una biblioteca o colección de código, además de estructuras y convenciones para crear una aplicación web. Técnicamente, es una biblioteca de paquetes (un RubyGem), que se instala utilizando la interfaz de línea de comandos del sistema operativo y añade

funcionalidad al lenguaje Ruby. Rails proporciona estructuras y convenciones para el desarrollo web, por lo que se requiere menos código para construir una aplicación web. Todas las aplicaciones de Rails utilizan las mismas estructuras, proporcionando consistencia entre las aplicaciones. Las convenciones comunes y bien conocidas facilitan que los desarrolladores colaboren en las aplicaciones de Rails y compartan mejoras y bibliotecas de código adicionales con una comunidad amplia. Las estructuras y convenciones de Rails están codificadas como la API Rails (la interfaz de programación de aplicaciones, o directivas que controlan el código). La API de Rails se documenta en línea y se describe en libros, artículos y publicaciones de blog.

8.1.8 JavaScript

Según Santos García y Sánchez García (citado por Osuma), es un lenguaje de programación interpretado normalmente por los navegadores web; por lo tanto, puede correr del lado del cliente. Es usado para darle capacidades dinámicas a los contenidos HTML, como por ejemplo, menús de contenidos, ventanas emergentes, validaciones de datos introducidos por el usuario, entre muchas otras funcionalidades.

8.1.9 JQuery

JQuery es una biblioteca de JavaScript rápida, pequeña y característica. Hace cosas como el desplazamiento y manipulación de documentos HTML, manejo de eventos, animación y Ajax mucho más simple con una API fácil de usar que funciona a través de una multitud de navegadores.(The jQuery Foundation, 2017).

8.1.10 Firebase

Firebase Cloud Messaging es un poderoso sistema que ya admite el envío de mensajes a apps de iOS y Android, y a Chrome. Los desarrolladores podrán usar FCM para enviar mensajes a navegadores que admitan la Push API, lo cual te permitirá expandirte más allá de Chrome y también realizar envíos a Firefox, Opera y otros.

Actualmente, la biblioteca de JavaScript de FCM permite a los desarrolladores acceder a navegadores compatibles con la Push API. A saber:

- Chrome para escritorio y dispositivos móviles (versiones 50 y posteriores)
- Firefox para escritorio y dispositivos móviles (versiones 44 y posteriores)
- Opera Mobile (versiones 37 y posteriores)

www.bdigital.ula.ve

Apéndice 2

Guías y manuales de uso para el usuario final.

GeRAIS en su versión 3.0 es un sistema que permite el manejo y gestión de un curso usando la estrategia RAIS como método de enseñanza.

9.1 Instalación de la aplicación.

Para la instalación es necesario contar con el siguiente ambiente:

- Ruby y Ruby On Rails
- MySQL
- Git (para descargar desde el control de versiones remoto).

9.1.1 Pasos de instalación.

Mediante el uso de la herramienta para el control de versiones Git descargue una copia del proyecto ubicado en la siguiente dirección <https://github.com/yoelero/graais>. El framework Ruby on Rails proporciona una serie de comandos que se pueden ejecutar en la línea de comandos permitiendo crear una nueva aplicación, crear una base de datos, ejecutar un servidor local, entre otros.

Los pasos para ejecutar el proyecto son los siguientes:

1. A través de la consola ejecutar el comando `git clone` para hacer checkout al repositorio remoto, `git clone https://github.com/yoelero/graais.git`, esto creará una carpeta con el nombre `graais`.
2. Crear la base de datos con el manejador de base de datos.

3. Por medio de consola ubicarse dentro del proyecto y ejecutar el comando `bundle install` para agregar al proyecto las gemas especificadas en el archivo Gemfile.
4. Situarse dentro de la carpeta del proyecto y modificar el archivo `database.yml`, ubicado en `grais/config`, con los parámetros correctos para conectarse a la base de datos.
5. Por medio de consola ubicarse dentro del proyecto y ejecutar el comando `rails db:migrate` para crear las tablas de la base de datos.
6. Abrir una nueva consola para correr el servidor local que proporciona Rails, ubicarse dentro del proyecto y ejecutar el comando `rails server` o simplemente `rails s`.
7. Abrir el navegador y escribir la dirección `localhost:3000`.

9.2 Requisitos para su uso.

Nota: Ya que la arquitectura de GeRAIS es web, por lo tanto se requiere que exista previamente un servidor accesible corriendo la aplicación.

Para conectarse a la aplicación GeRAIS como cliente es necesario los siguientes requisitos:

- Tener algún dispositivo desde el cual se pueda conectar a la aplicación, ya sea un Smartphone, una tablet o un computador.
- Tener un navegador web instalado con soporte para JavaScript (La mayoría de los navegadores lo tienen), como Firefox, Chrome, Opera, Safari, Explorer entre otros.
- Estar en una red que tenga acceso al servidor de GeRAIS.

9.3 Aspectos generales.

Al cargar el sistema en su navegador aparece la pantalla principal que se muestra a continuación:



Figura 14: Pantalla inicial



Figura 15: Pantalla inicial para
móvil y tables

En la pantalla inicial se da la bienvenida al usuario y se presenta un formulario para iniciar sesión y enlace directo para registrarse. El diseño adaptativo móvil y tables muestra un icono de menú que contiene enlaces directos a iniciar sesión y registrarse.

9.4 Guía del usuario general.

9.4.1 Aspecto general.

Para todas las acciones siguientes se necesita que el usuario haya iniciado una sesión en el sistema.

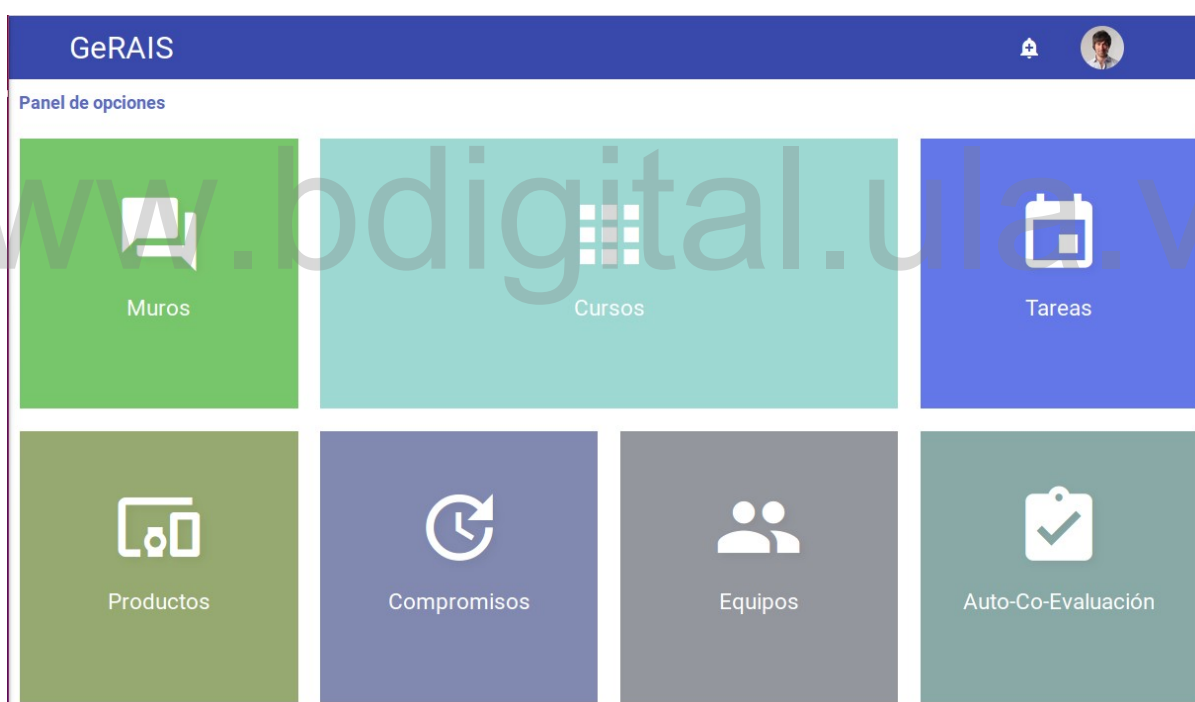


Figura 16: Pantalla de panel de opciones

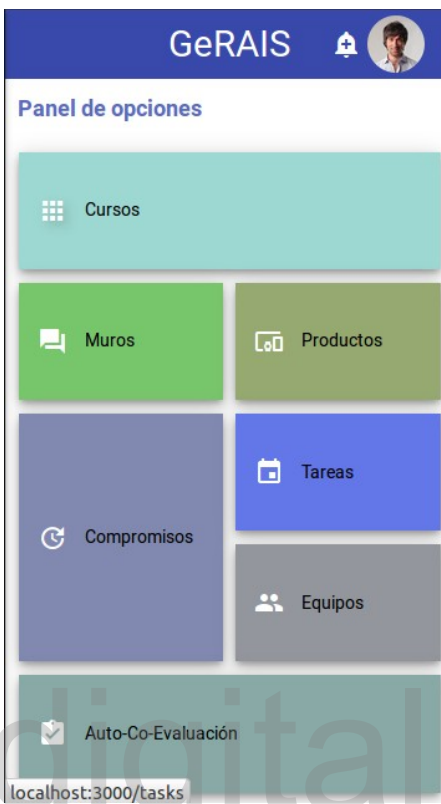


Figura 17: Pantalla de panel de opciones para móvil y table.

En el cuerpo dicha pantalla podemos encontrar enlaces directos a los componentes del sistema.

Nota: Para todas las vistas de la aplicación en el encabezado se encuentran enlaces directos a notificaciones (icono "add alert"), así como perfil del usuario y cerrar sesión (imagen del usuario).

9.4.2 Editar perfil.

Pulsar sobre la fotografía en el encabezado de la página y luego sobre el enlace a perfil. Una vez en la página pulsar sobre editar perfil, seguidamente desplegará un formulario para completar y luego pulsar guardar. De la misma forma para editar la contraseña.



Figura 18: Pantalla perfil de usuario

9.4.3 Gestionar notificaciones.

Para acceder a notificaciones se debe pulsar sobre el icono “add alert” sobre la cabecera de la página. Si la notificación se trata de una solicitud de acceso a un curso o equipo, se mostrará el botón aceptar. Para eliminar la notificación pulse sobre el icono “close” y luego confirmar.

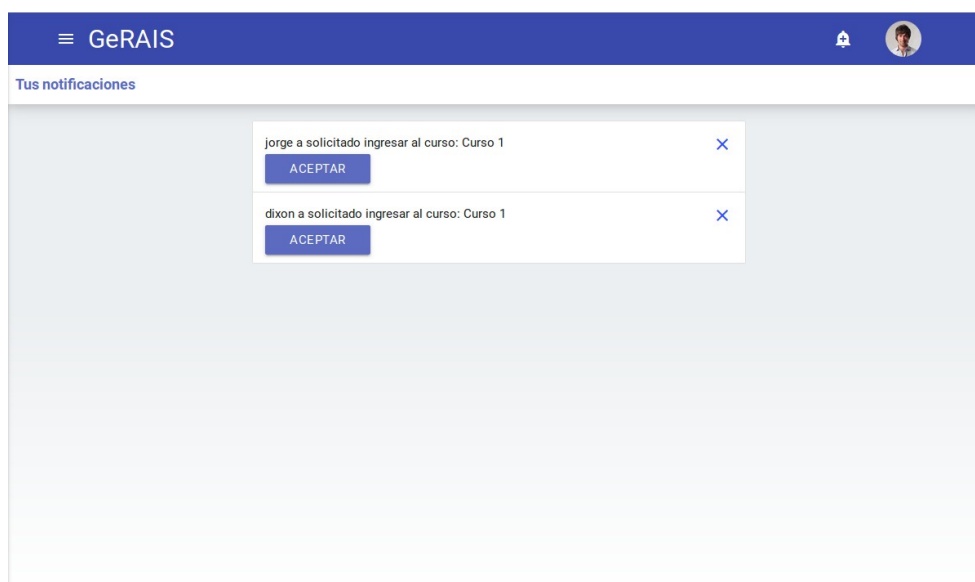


Figura 19: Pantallas de notificaciones de usuario.

9.4.4 Registrar un nuevo curso.

Para registrar un curso en usuario de ubicarse en cursos, a través del enlace presentado en el panel de opciones. Presionar sobre el icono “add” que desplegara un formulario para completar y luego pulsar el botón de guardar.



Figura 20: Pantalla para crear curso.

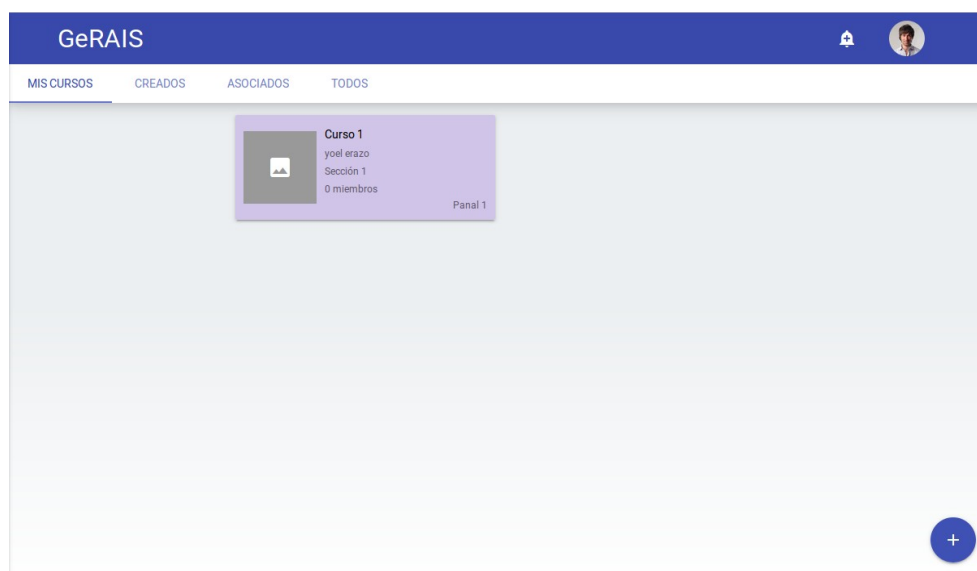


Figura 21: Pantalla para crear curso alternativo.

Nota: Si el usuario no forma parte de ningún curso se mostrará un mensaje y un icono “add”. Si forma parte de algún curso en la parte inferior de la pantalla se mostrara el icono “add”.

9.4.5 Solicitar unirse a un curso.

Requisitos previos:

- El usuario no debe formar parte del curso.

Ubicarse dentro del curso al cual desea pertenecer, pulsar el botón con el icono “check”, ubicado sobre la barra lateral, en la versión web móvil se ubica sobre la imagen parallax.

Nota: todos los cursos registrados en la herramienta se pueden encontrar en la sección todos del ambiente de cursos.

9.4.6 Ingresar al curso a través de código.

Requisitos previos:

- El usuario no debe formar parte del curso.

Ubicase dentro del curso al cual desea pertenecer, pulsar el botón con el icono “check”, ubicado sobre la barra lateral, en la versión web móvil se ubica sobre la imagen parallax. Se desplegará un formulario para introducir código y luego pulsar listo.

Nota: todos los cursos registrados en la herramienta se pueden encontrar en la sección **todos** del ambiente de cursos.

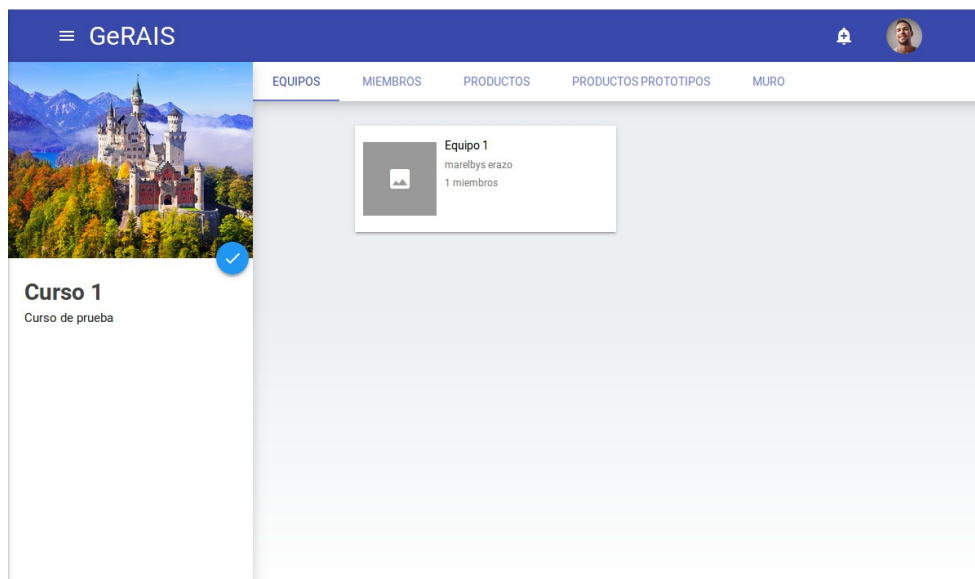


Figura 22: Pantalla curso solicitud.

9.5 Guía del profesor.

Al crear un curso el usuario pasa a ser automáticamente el profesor o jefe ejecutivo del curso.

9.5.1 Editar curso.

Ubicarse dentro del curso en el cual se desempeña como jefe ejecutivo, pulsar sobre el botón de con el icono editar, se desplegará un formulario con los datos a modificar y luego pulsar guardar.

9.5.2 Eliminar curso.

Requisitos previos:

- El curso no debe tener miembros o productos prototipos.

Ubicarse dentro del curso en el cual se desempeña como jefe ejecutivo, pulsar el botón con el icono cerrar y luego confirmar su eliminación.

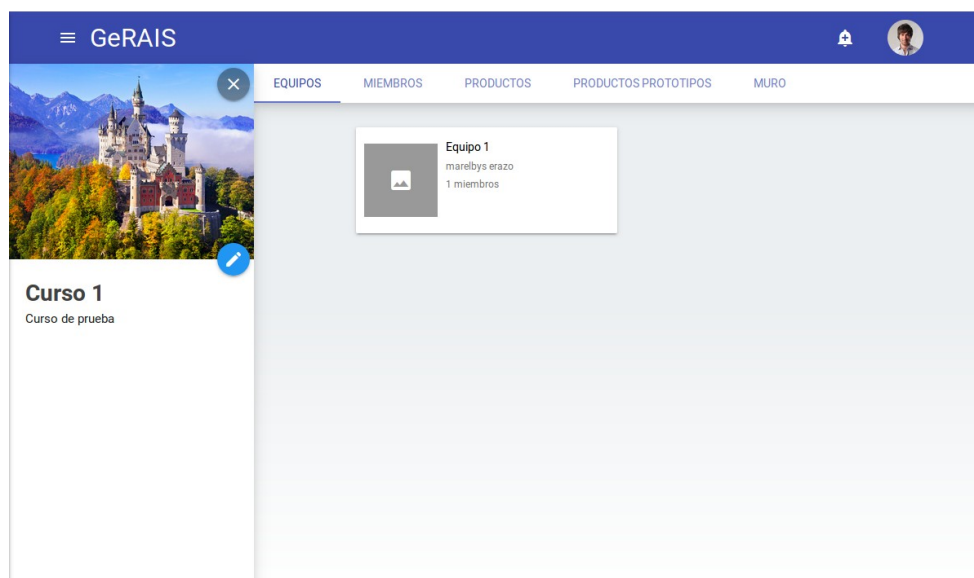


Figura 23: Pantalla curso para jefe ejecutivo.

9.5.3 Editar porcentaje de avance de la tarea.

Elegir la tarea a la cual desea editar porcentaje de de avance. Desplazar la barra de avance para establecer un porcentaje.

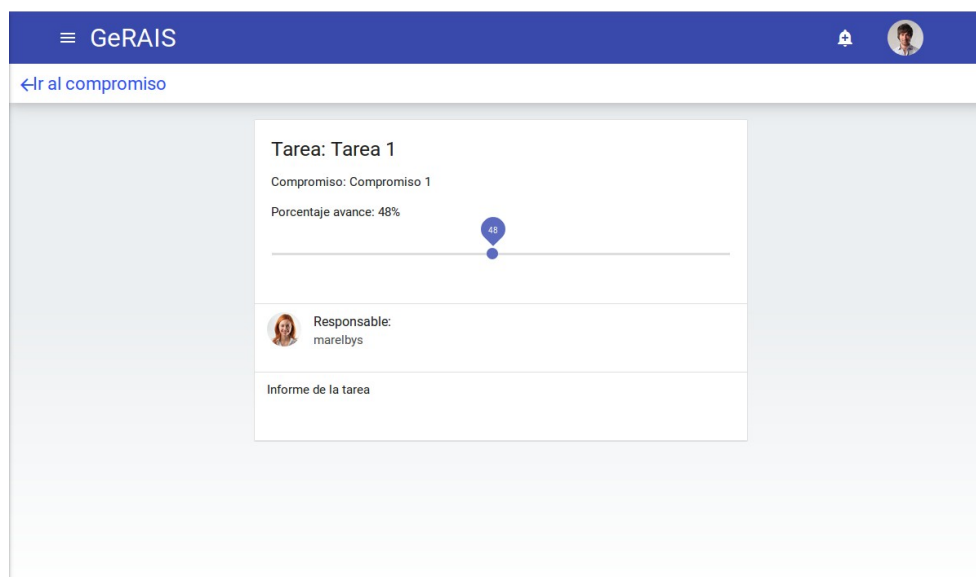


Figura 24: Pantalla tarea para el jefe ejecutivo.

9.5.4 Editar fecha del compromiso.

Seleccionar un compromiso, pulsar sobre botón de editar en la parte inferior de la pantalla. Se desplegará un formulario para asignar una nueva fecha y luego pulsar guardar.

Nota: Solo el jefe ejecutivo puede editar la fecha de los compromisos ya creados.

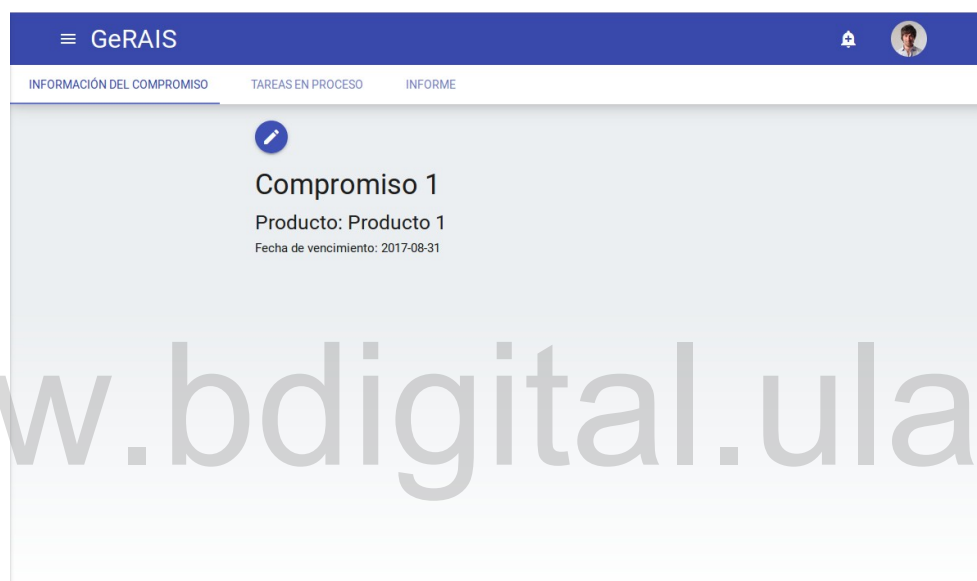


Figura 25: Pantalla compromiso para profesor.

9.6 Guía del estudiante.

Los usuarios que accedan al curso a través de una solicitud o código pasan a ser estudiantes del curso.

9.6.1 Retirarse del curso.

Ingresar al curso en el cual se desempeña como estudiante. Pulsar sobre el botón “Retirarse del curso” ubicado en la barra lateral, si se trata de la versión web para móvil dirigirse a la sección información.

9.6.2 Crear un equipo.

Requisitos previos:

- El miembro del curso no debe pertenecer a ningún equipo.

Presionar sobre el botón con icono “add”, se desplegará un formulario para introducir datos y luego presionar guardar.

Nota: El usuario que crea el equipo automáticamente pasa a desempeñar el rol de líder.

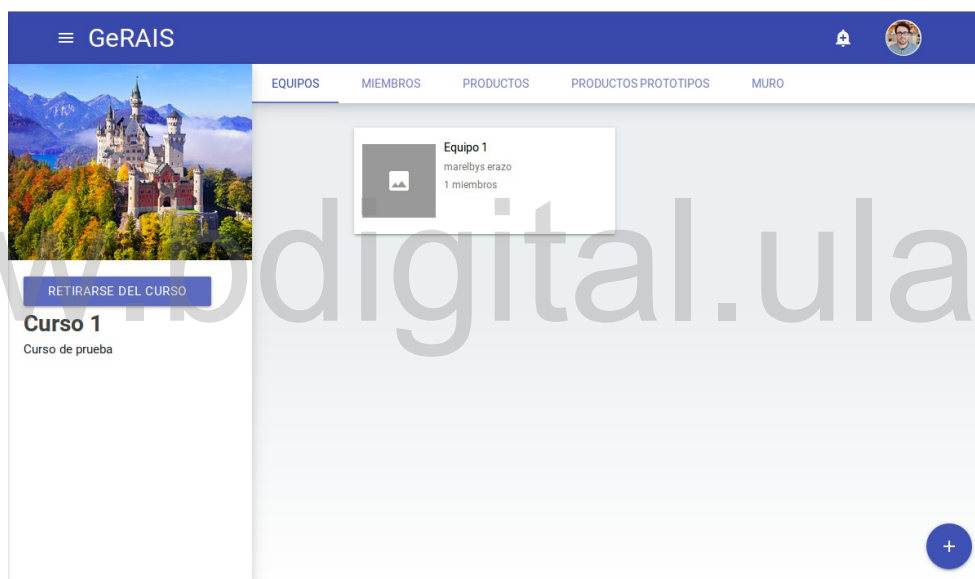


Figura 26: Pantalla curso para estudiante sin equipo.

9.6.3 Solicitud para unirse a un equipo.

Ubicarse dentro del equipo al cual desea pertenecer, presionar botón con icono “check”, ubicado sobre la barra lateral, en la versión web móvil se ubica sobre la imagen parallax.

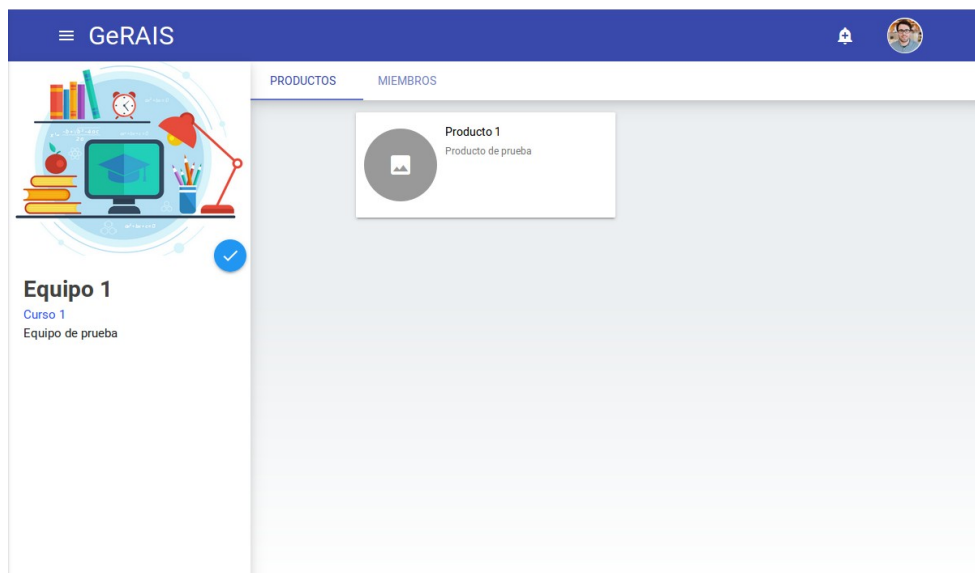


Figura 27: Pantalla para estudiante que no forma parte de un curso.

9.6.4 Editar equipo.

Ubicarse en el equipo en el cual se desempeña como líder, presionar el botón de editar con icono “mode edit” ubicado en la barra lateral, se desplegará un formulario modificar los datos y pulsar guardar. En la versión web para móvil encontrará el botón en la imagen parallax.

9.6.5 Eliminar equipo.

Requisitos previos:

- El equipo no debe tener integrantes o productos.

Ubicarse en el equipo en el cual se desempeña como líder, presionar el botón con icono “close” ubicado en la parte superior de la barra lateral, en la versión web para móvil está ubicado sobre la imagen paraxas, y luego confirmar.

9.6.6 Crear un producto.

Requisitos previos:

- El usuario debe ser líder del equipo.

Situarse dentro del equipo, la sección productos siempre está seleccionada por omisión, presionar sobre el botón con icono “add”, se desplegará dos botones uno para crear un producto y otro para heredar un producto prototipo. Si selecciona crear se desplegará un formulario para llenar los datos y luego presionar guardar, si por el contrario desea heredar de un producto prototipo se mostrará una lista con los productos prototipos.

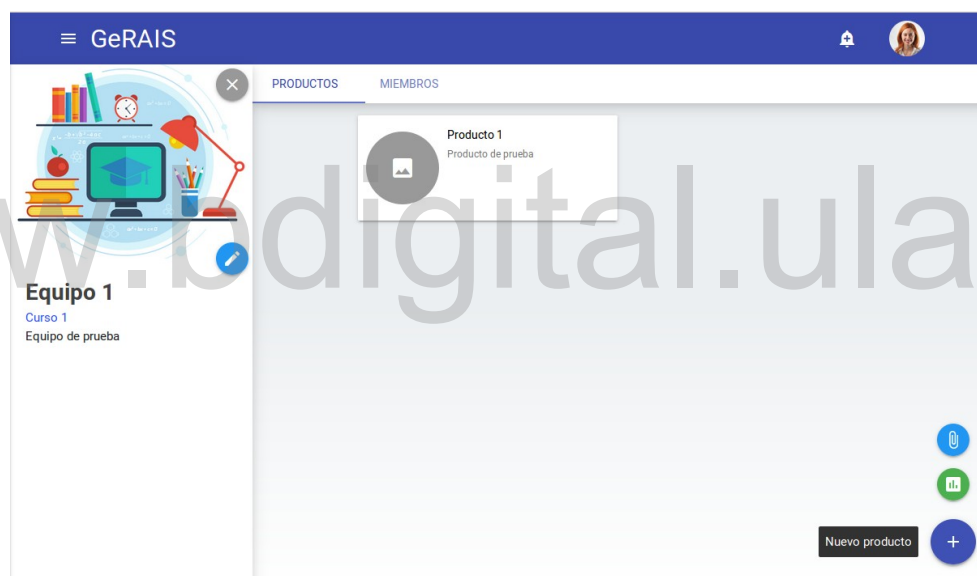


Figura 28: Pantalla de equipo para estudiante líder

9.6.7 Editar un producto.

Requisitos previos:

- El usuario debe ser líder del equipo.

Situarse dentro del producto, presionar el botón de editar con icono “mode edit” que se encuentra ubicado sobre la barra lateral, en la versión web móvil se encuentra sobre la

imagen parallax, se desplegará un formulario con los datos a modificar y luego pulsar guardar.

Nota: Si se trata de un producto prototipo no se puede editar.

9.6.8 Eliminar un producto.

Requisitos previos:

- No deben haber compromisos.

Situarse dentro del producto, presionar sobre icono “close” que se encuentra en la parte superior de la barra lateral, en la versión web móvil se encuentra sobre la imagen parallax, y luego confirmar.

Nota: Si se trata de un producto prototipo no podrá eliminar el producto.

9.6.9 Crear compromiso.

Requisitos previos:

- El usuario debe ser líder del equipo.

Ubicarse sobre el producto, la sección de compromisos en proceso siempre está seleccionada por omisión, presionar sobre el botón con icono “add”, se desplegará un formulario para introducir los datos y luego presionar guardar.

Nota: si se trata de un producto prototipo no podrá crear compromisos.

9.6.10 Crear informe del producto.

Requisitos previos:

- El usuario debe ser líder del equipo.

Ubicarse sobre el producto, seleccionar la sección informe, automáticamente mostrará un formulario para redactar informe y luego presionar guardar

9.6.11 Editar informe del producto.

Requisitos previos:

- El usuario debe ser líder del equipo.

Ubicarse sobre el producto, seleccionar la sección informe, presionar icono de editar, se desplegará un formulario con los datos a modificar y luego presionar guardar.

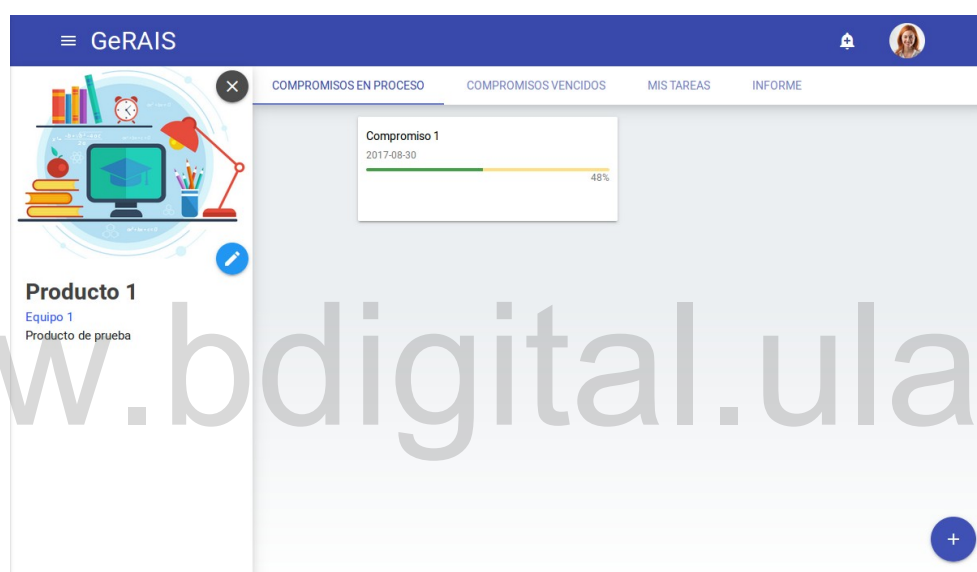


Figura 29: Pantalla de producto para estudiante líder

9.6.12 Editar compromiso.

Requisitos previos:

- El usuario debe ser líder del equipo.

Ubicarse sobre el compromiso en la sección de información de compromiso, pulsar el botón con icono “mode edit”, se desplegará un formulario con los datos a modificar y luego pulsar guardar.

Nota: no podrá editar compromiso si se trata de un compromiso prototipo.

9.6.13 Eliminar compromiso.

Requisitos previos:

- El usuario debe ser líder del equipo.
- No deben haber tareas.

Ubicarse sobre el compromiso en la sección de información de compromiso, presionar el botón con icono “close” y luego confirmar.

Nota: si se trata de un compromiso prototipo no se podrá eliminar.

9.6.14 Crear informe del compromiso.

Requisitos previos:

- El usuario debe ser líder del equipo.

Ubicarse en el compromiso, seleccionar la sección informe, automáticamente mostrará un formulario para redactar informe y luego presionar guardar

9.6.15 Editar informe del compromiso.

Requisitos previos:

- El usuario debe ser líder del equipo.

Ubicarse en el compromiso, seleccionar la sección informe, presionar botón de editar con icono “mode edit”, se desplegará un formulario con los datos a modificar y luego presionar guardar.

9.6.16 Crear tarea.

Requisitos previos:

- El usuario debe ser líder del equipo.

Ubicarse en el compromiso, la sección de tareas siempre está seleccionada por omisión, presionar sobre el botón con el icono “add”, se desplegará un formulario para llenar y luego pulsar guardar.



Figura 30: Pantalla compromiso para estudiante líder

9.6.17 Editar tarea.

Requisitos previos:

- El usuario debe ser líder del equipo.

Ubicarse en la tarea, pulsar botón editar con icono “mode edit”, ubicado en la parte superior del cuerpo de la página, se desplegará un formulario con los datos a modificar y luego presionar guardar.

9.6.18 Eliminar tarea.

Requisitos previos:

- El usuario debe ser líder del equipo.
- No debe haber redactado el informe de la tarea.

Ubicarse en la tarea, pulsar botón eliminar con icono “close”, ubicado en la parte superior del cuerpo de la pagina y luego confirmar.

9.6.19 Crear informe de tarea.

Requisitos previos:

- El usuario debe ser el responsable de la tarea.

Ubicarse en la tarea, se mostrará automáticamente un formulario para crear la tarea y luego presionar guardar.

9.6.20 Editar informe de tarea.

Requisitos previos:

- El usuario debe ser el responsable de la tarea.

Ubicarse en la tarea, presionar sobre el botón editar, se desplegará un formulario con los datos a modificar y luego guardar.

9.6.21 Eliminar informe de tarea.

Requisitos previos:

- El usuario debe ser el responsable de la tarea.

Ubicarse en la tarea, presionar sobre el botón eliminar con icono “close” y luego confirmar.

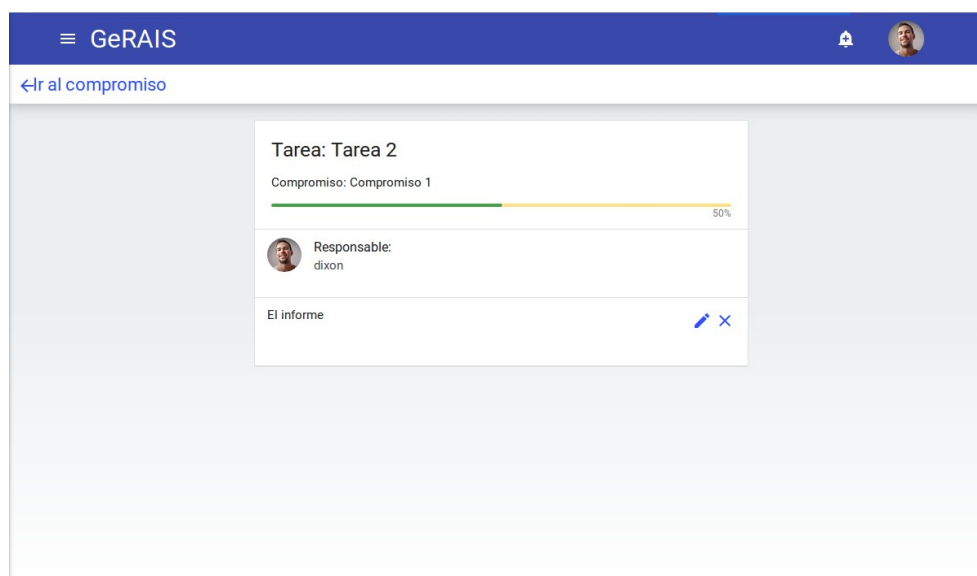


Figura 31: Pantalla tarea estudiante responsable de la tarea.

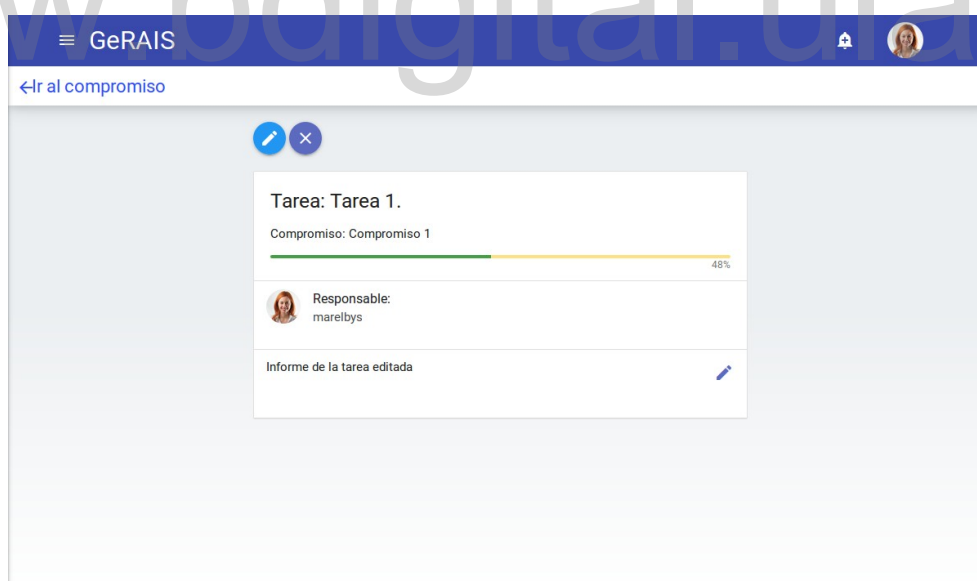


Figura 32: Pantalla de tarea para estudiante líder.

9.7 Muro del curso

Cada curso cuenta con un canal de comunicación denominado muro. Solo los miembros del curso pueden crear comentarios, así como editarlos y eliminarlos.

9.7.1 Crear un comentario.

Ubicarse en el muro de un curso, por defecto mostrará un formulario para llenar y luego guardar. También es posible crear comentarios a un comentario, por defecto se muestra un formulario para llenar y luego guardar.

9.7.2 Editar un comentario.

Requisitos previos:

- Solo quien crea el comentario puede editarlo.

Ubicarse en el muro de un curso, elegir el comentario pulsar el icono “mode edit”, se desplegará un formulario con los datos a modificar y luego guardar.

9.7.1 Editar un comentario.

Requisitos previos:

- Solo quien crea el comentario puede eliminarlo.

Ubicarse en el muro de un curso, elegir el comentario pulsar el icono “close” y luego confirmar.

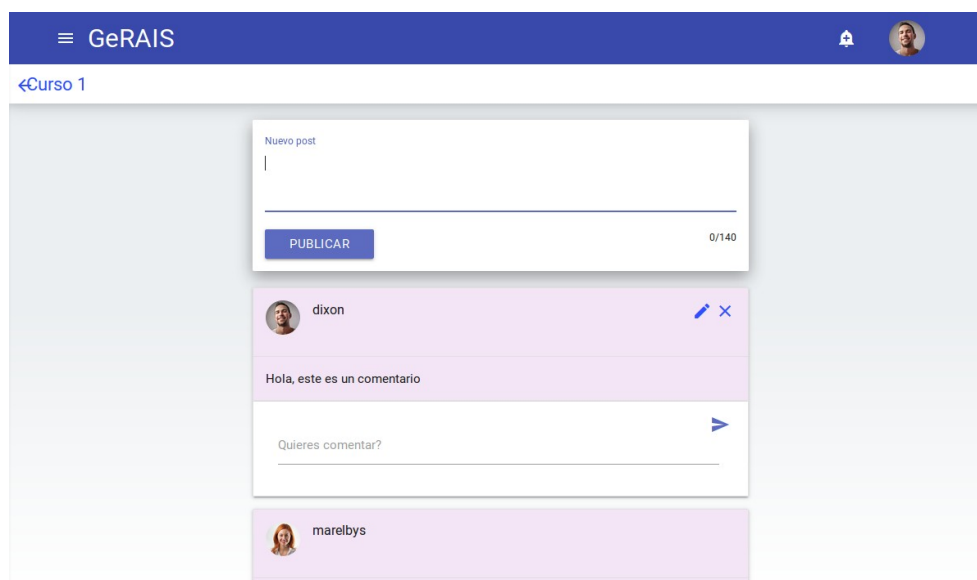


Figura 33: Pantalla muro para miembros del curso.

www.bdigital.ula.ve