



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

MERIT-ON: SISTEMA DE RECOMPENSAS PARA ESTUDIANTES APLICADOS EN CENTROS TIC

Alumno: Antonio Estrella Cruz

Tutor: Prof. D. Jose María Serrano Chica

Dpto: Departamento de Informática

Octubre, 2019



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don JOSE MARÍA SERRANO CHICA , tutor del Proyecto Fin de Carrera titulado: MERIT-ON: SISTEMA DE RECOMPENSAS PARA ESTUDIANTES APLICADOS EN CENTROS TIC, que presenta ANTONIO ESTRELLA CRUZ, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, OCTUBRE de 2019

El alumno:

Los tutores:

ANTONIO ESTRELLA CRUZ

JOSE MARÍA SERRANO CHICA

Índice

1. INTRODUCCIÓN	5
1.1. Introducción al proyecto	5
1.2. Motivación	6
1.3. Propósito del proyecto.....	7
1.3.1. Objetivos	7
1.3.2. Requisitos y funcionalidades de la aplicación.....	8
1.4. Estructura del documento	9
2. ANÁLISIS.....	10
2.1. Análisis preliminar	10
2.2. Propuesta de solución.....	11
2.3. Planificación	11
2.3.1. Diagrama de Gantt.....	11
2.3.2. Estimación de costes	12
2.4. Resumen del Análisis.....	14
3. EL ESTADO DEL ARTE	15
3.1. Metodología	15
3.1.1. Modelo en Cascada	15
3.1.2. Modelo de prototipos.....	17
3.1.2.1. Etapas.....	17
3.1.3. Modelo en espiral.....	17
3.1.4. Modelo Incremental.....	18
3.2. Arquitectura.....	20
3.3. Frameworks	22
3.3.1. Codeigniter.....	22
3.3.2. Zend.....	22
3.3.3. Phalcon	23
3.3.4. CakePHP	24
3.4. Elección de Codeigniter	24
4. TECNOLOGÍAS UTILIZADAS	25
4.1. PHP	25
4.2. JavaScript	26
4.3. HTML5	26
4.4. CSS	27
4.5. Bootstrap.....	28
4.6. XAMPP	28

4.7.	Codeigniter.....	29
4.8.	Visual Paradigm.....	29
4.9.	GanttProject.....	29
4.10.	Conclusión acerca de las tecnologías utilizadas.....	30
5.	DESARROLLO.....	31
5.1.	Primer incremento.....	31
5.1.1.	Análisis.....	31
5.1.2.	Diseño.....	34
5.1.3.	Implementación.....	45
5.1.4.	Pruebas.....	48
5.2.	Segundo Incremento.....	49
5.2.1.	Análisis.....	49
5.2.2.	Diseño.....	50
5.2.3.	Implementación.....	51
5.2.4.	Pruebas.....	52
5.3.	Tercer incremento.....	53
5.3.1.	Análisis.....	54
5.3.2.	Diseño.....	55
5.3.3.	Implementación.....	58
5.3.4.	Pruebas.....	60
5.4.	Cuarto Incremento.....	61
5.4.1.	Análisis.....	61
5.4.2.	Diseño.....	62
5.4.3.	Implementación.....	63
5.4.4.	Pruebas.....	65
5.5.	Revisión del prototipado.....	66
6.	CONCLUSIONES Y MEJORAS FUTURAS.....	67
	ANEXO I. MANUAL DE INSTALACIÓN.....	69
	ANEXO II. MANUAL DE USUARIO.....	71
	BIBLIOGRAFÍA.....	78

1. INTRODUCCIÓN

Este proyecto trata de integrar el concepto de ludificación dentro de las áreas más básicas de la educación. En concreto, crear un sistema de medallas, basado en el diseño y mecánicas de los sistemas de puntuación como los utilizados en los videojuegos, buscando la atención, el aprendizaje, la diversión y sobre todo su motivación personal.

Mediante una breve introducción al proyecto, alegaré los motivos por los cuales me he decantado por esta elección, en qué objetivos se va a basar la obtención de medallas, y por último los objetivos que satisface esta aplicación.

1.1. Introducción al proyecto

La meritocracia es el sistema educativo por excelencia en la sociedad actual. Esta meritocracia no está considerada como pura, ya que existen factores sociales, políticos y económicos que la alteran.

Según la RAE, el sistema meritocrático es un sistema de gobierno en que los puestos de responsabilidad se adjudican en función de los méritos personales. Como influyen otros factores mencionados anteriormente, este sistema no es equitativo. Por ello con la creación de esta aplicación se busca generar un sistema equitativo basado en la obtención de medallas por parte de los alumnos. Dichas medallas serán asignadas a los alumnos mediante el profesor.

A través de la investigación de estudios psicológicos y pedagógicos, es posible determinar el tipo de comportamiento y objetivos que tienen que desarrollar los alumnos, e implementar las medallas, según esos factores, para desarrollar un producto lo más equitativo e inclusivo posible. Lo que se pretende desarrollar es un sistema en el que todos los alumnos tengan las mismas posibilidades de obtener medallas, para que al final del curso, haya una integración positiva, y juntos hayan cumplido unos objetivos globales e individuales.

En las primeras etapas de la educación, se suele hacer uso de los sistemas de insignias y medallas cuando el alumno alcanza algún objetivo, lo cual significa para ellos obtener una recompensa. No obstante, en las siguientes etapas, este sistema de

reconocimiento suele ser considerado infantil y poco estimulante para estudiantes de niveles superiores. Esta tendencia está cambiando, debido a la integración de procesos de ludificación, por ejemplo, en instituciones como el MIT, o grandes empresas como Pixar, emiten insignias digitales como símbolo de reconocimiento.

El proyecto a desarrollar se basa en la observación e investigación de una muestra de alumnos en distintos centros educativos, con la consecuente obtención de unos resultados que ayudarán al desenvolvimiento de la creación de medallas o insignias, apoyándose en los patrones comunes encontrados en los distintos centros, y valorando positivamente los menos comunes pero que tengan una importante carga psicológica.

1.2. Motivación

Hoy en día la educación que se está llevando a cabo en la sociedad actual, cada vez está más ligada con la tecnología. No obstante, con el rápido avance de la misma, la evolución que se ha sufrido en la educación ha sido torpe, y dañina. La forma principal de impartir clases mediante el uso de la tecnología en las aulas se trata de un modelo único, en el que hay pocas oportunidades para que los alumnos se involucren (p. ej. leer diapositivas). Esta limitación crea un entorno en el cual el alumno se ve incapaz de involucrarse, tanto en el aprendizaje, como en la formación de vínculos con el profesor y otros alumnos. Para acabar con estas limitaciones, es necesario considerar una forma de educar multimodal.

Un enfoque multimodal como el sistema de medallas creado para este proyecto, incluye varios de los sentidos humanos cuando el alumno interactúa con el sistema y con otros alumnos, lo que significa que los alumnos se sumergen y se interesan por el aprendizaje de una manera más activa.

Hay algunas plataformas que ya están en el mercado (p. ej. ClassDojo), las cuales usan una gran variedad de herramientas con un enfoque multimodal que permiten a los profesores diseñar estrategias educativas que tienen el potencial de sumergir a los alumnos en una experiencia de aprendizaje más inclusiva y motivadora, lo que supone un incremento en el desarrollo educativo y en los resultados académicos.

El principal propósito de este proyecto es crear un sistema de medallas que mejore el enfoque unimodal que se está usando ahora mismo en todos los niveles de educación obligatoria.

Otro factor que ha motivado el desarrollo de este proyecto, es mi experiencia laboral, mi espíritu emprendedor, y sobre todo el carácter de beneficio social que tiene este proyecto detrás. El hecho de poder desarrollar una aplicación, que puede tener un beneficio muy positivo de cara al futuro, ya que está orientada a la base de la civilización actual, la educación de las generaciones venideras con un enfoque multimodal, en el cual no sólo se desarrolle la competencia, sino la cooperación, la empatía y la superación. A su vez, creo que puede ser una oportunidad para hacer un proyecto global, estudiando y adaptando la aplicación a la educación que se lleva a cabo en cada región del mundo.

1.3. Propósito del proyecto

El proyecto consiste en primer lugar en el estudio de las plataformas ya existentes de gamificación tanto educativas como empresariales, y de la necesidad tecnológica en la educación que hay actualmente. Una vez sean analizados los recursos y las necesidades, se seleccionarán los que nos ayuden a crear una aplicación que mejore el sistema actual.

1.3.1. Objetivos

Los objetivos de los que partimos para la realización del proyecto son:

- Tomar una muestra de una serie de centros educativos, para ver sus carencias, y los puntos comunes.
- Análisis de los principales frameworks para el desarrollo de la aplicación, y el porqué de la elección de uno de ellos en detrimento de los otros.
- Análisis de los diferentes patrones de arquitectura software, eligiendo y justificando la elección de uno de ellos para el desarrollo de la aplicación web.
- Desarrollar la aplicación web, para su accesibilidad desde distintos dispositivos de manera responsiva.

1.3.2. Requisitos y funcionalidades de la aplicación

La aplicación constará de una serie de opciones y requisitos los cuales se muestran a continuación:

- Personalización del sistema de medallas, mediante el uso de editores de imágenes.
- Añadir un sistema de creación de usuarios en el panel de administrador.
- Proporcionar seguridad en las contraseñas.
- Proporcionar seguridad en las vistas, para que no pueda acceder un usuario no registrado.
- Proporcionar seguridad en las vistas, para que, dependiendo del tipo de usuario, pueda acceder solamente a las suyas en cuestión.
- Crear distintas vistas dependiendo del usuario en cuestión.
- Justificación de la asignación de tutores legales con el alumnado y profesorado con el alumnado.
- Desarrollar el sistema de creación de medallas.
- Añadir una tabla con los datos de los alumnos y la opción de asignación de las medallas.
- Desarrollar el sistema de visión de medallas asignadas por parte de los profesores para los tutores legales.
- Desarrollar el sistema de visión de medallas asignadas positivas por parte de los profesores para los alumnos.
- Crear las vistas, controladores y modelos de cada tipo de usuario.
- Crear usuarios de prueba por cada tipo de usuario que exista.
- Elaborar una memoria donde se recoja todo el trabajo desarrollado.
- Incluir en la memoria las distintas pruebas realizadas que certifiquen el correcto funcionamiento de la aplicación.
 - Pruebas de asignación de alumnos a tutores legales y profesores.
 - Pruebas de asignación de medallas a los alumnos.
 - Pruebas de visualización de las distintas medallas por parte de los alumnos y de los tutores legales.

1.4. Estructura del documento

A continuación, se va a indicar como se ha estructurado y organizado la memoria y se proporciona una breve descripción de cada apartado:

- 1. Introducción: en este apartado se muestran las razones por las que se ha realizado este trabajo, así como los objetivos necesarios del proyecto y un pequeño desglose de la estructura del mismo.
- 2. Análisis: en este capítulo se realiza un análisis preliminar del proyecto, en el que se han recopilado opiniones de fuentes externas. Se ha realizado la planificación del proyecto, y un apartado en el que se muestran los costes que se han utilizado para el mismo.
- 3. Estado del arte: aquí se realiza una comparativa entre las distintas metodologías que existen para realizar el proyecto y cual de ellas se ha elegido, aparte se realiza otra comparativa entre distintos frameworks y se realiza la elección de uno de ellos.
- 4. Tecnologías utilizadas: en este apartado, se hace un breve resumen y descripción de todas las tecnologías que se han utilizado para el desarrollo del proyecto.
- 5. Desarrollo: a partir de este capítulo se realiza la explicación de la implementación de las distintas fases del modelo seleccionado en el capítulo 3.
- 6. Conclusiones y mejoras: en este capítulo se exponen las impresiones obtenidas una vez terminado el trabajo, y un conjunto de mejoras a realizar sobre el prototipo.
- Anexos: el primero de los dos anexos incluidos está formado por el manual de instalación de la aplicación. La segunda parte muestra un manual de usuario.
- Bibliografía: desglose de las referencias que se han utilizado para la realización de la memoria.

2. ANÁLISIS

2.1. Análisis preliminar

El proyecto desarrollado consiste en una aplicación orientada al sistema educativo obligatorio. Se parte de una observación de dicho sistema actual, en el que se han podido observar carencias a nivel tecnológico, ya sea por la escasa formación docente en temas de nuevas tecnologías, o por la falta de medios.

Durante nuestra historia reciente se ha intentado aplicar la tecnología en todos los ámbitos de la vida, habiéndose llevado a cabo con más o menos éxito en según el campo donde se haya puesto en marcha. En el caso de la educación, podemos hablar de que ha tenido un éxito “relativo”. La implementación de la tecnología en este campo, ha sido en una parte fundamental deficiente, por ejemplo, con la implantación de plataformas internas, con una curva de aprendizaje elevada para el desconocimiento de la tecnología por parte del profesorado, poco eficientes, y poco atractivas visualmente. Actualmente esta tendencia está cambiando, debido a la renovación de edad por parte de los equipos docentes de cada centro, por este motivo, son más propensos a entender y a abrir la mente a nuevos sistemas.

Otra de las problemáticas que se han encontrado durante el análisis es la forma en la que el sistema de puntuaciones ha sido llevado a cabo a lo largo de la historia. Se trata de un sistema anticuado, en el que se valora con puntuaciones, y se discrimina involuntariamente, pero con un efecto negativo en el alumnado.

En la actualidad, existen plataformas que abordan estos problemas, p.ej. ClassDojo es una de ellas. Es una aplicación que, mediante un sistema de puntuaciones positivas y negativas, motiva a los alumnos y mantiene informado a los tutores legales de la progresión de los niños. El enfoque utilizado para el desarrollo del proyecto es similar, se trata de una aplicación en la que los alumnos obtienen medallas según su desarrollo durante el curso, este sistema está basado en la meritocracia.

2.2. Propuesta de solución

El sistema a desarrollar consiste en una plataforma interactiva entre profesores y alumnado, en centros donde la tecnología está implantada, pero no se hace un uso correcto de ella. Considerando que todos los alumnos cuentan con un dispositivo móvil, la aplicación permite al profesorado la creación de logros (en forma de medallas o insignias), y asociarlos con la consecución de los objetivos, con los que reconocer y premiar a los alumnos.

2.3. Planificación

La planificación del proyecto es clave, para poder ajustar el entorno de trabajo al poder realizar aproximaciones sobre los costes y tiempo que va a requerir el desarrollo del proyecto.

2.3.1. Diagrama de Gantt

El diagrama de Gantt es una herramienta que permite conocer el tiempo que se estima para las diferentes actividades que se realizarán a lo largo del desarrollo del proyecto. El tiempo que durará cada actividad dependerá de su grado de dificultad, por lo que pueden llegar a durar horas, días o semanas. A raíz de la tabla expuesta a continuación obtendremos el diagrama de Gantt.

Tarea	Inicio	Duración	Final
Inicio de la planificación	04/01/2019	3 días	07/01/2019
Tomar una muestra de una serie de centros educativos	07/01/2019	2 días	09/01/2019
Incremento 1	09/01/2019	6 días	15/01/2019
Incremento 2	15/01/2019	8 días	23/01/2019
Incremento 3	23/01/2019	10 días	01/02/2019
Incremento 4	01/02/2019	12 días	12/02/2019
Pruebas	12/02/2019	5 días	17/02/2019

Redactar memoria	la	04/01/2019	59 días	03/03/2019
------------------	----	------------	---------	------------

Tabla 1. Tabla diagrama de Gantt

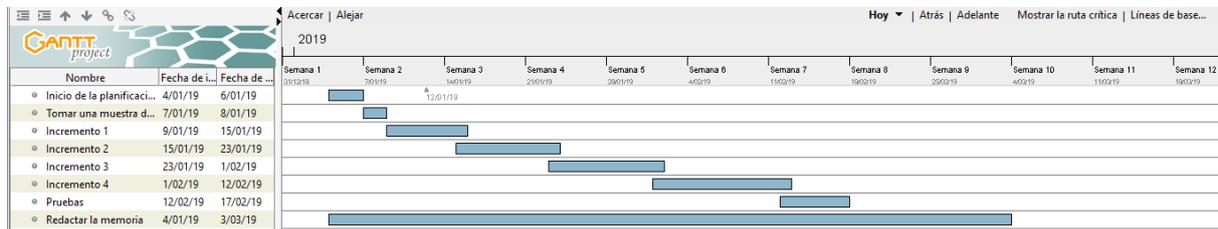


Ilustración 1. Diagrama de Gantt

2.3.2. Estimación de costes

Gracias a la estimación de los costes, se podrá definir con un margen de error ínfimo, cuánta inversión monetaria, va a necesitar la realización del proyecto, tanto la planificación y el diseño, como el propio desarrollo final.

Costes hardware

Componentes Hardware	Precio
Ordenador Portátil Asus K55VM	699 €

Tabla 2. Tabla de costes hardware

Costes software

Costes de las licencias del software utilizado para el desarrollo de la aplicación.

Componentes Software	Precio
Windows 10 Student Version	0€
Sublime text	0€
Xampp	0€
Google Chrome	0€
PHPMyAdmin	0€
Gitlab	0€
Microsoft Office 365	7€/mes x 3 meses = 21€

GanttProject	0€
Visual Paradigm	19€

Tabla 3. Tabla de costes software

Costes de personal

Estos costes están designados acorde al salario del personal que desarrolle el proyecto. Aunque el proyecto solamente lo haya realizado una persona, esta persona dispone de varios roles dentro del desarrollo del proyecto.

- Ingeniero de software: es el encargado de desarrollar el diseño y obtener los algoritmos necesarios para el desarrollo del proyecto.
- Programador: es el encargado de implementar el proyecto que ha diseñado el analista.

Puesto	Sueldo	Nº días	Nº horas	Total
Ingeniero de software	15€/hora.	11 días	88 horas	1320€
Programador	8€/hora	30 días	240 horas	1920€

Tabla 4. Tabla de costes de personal

Otros costes

Podríamos hablar de otros costes que han surgido durante la realización del proyecto, tales como la conexión a internet.

Concepto	Coste
Conexión a internet	40€/mes

Tabla 5. Tabla de costes variados

Coste Total

Coste	Precio
Coste Hardware	699€
Coste Software	40€
Costes de Personal	3240€
Otros Costes	120€
Coste Total	4099€

Tabla 6. Tabla de coste total

2.4. Resumen del Análisis

En primer lugar, se ha realizado un análisis sobre las carencias tecnológicas con las que cuentan los centros educativos, para el cual se ha propuesto una solución. Dicha solución ha sido planificada incluyendo los costes con los que va a contar el desarrollo. A continuación se procede a detallar la metodología y tecnologías que se han usado para la ejecución de dicha solución.

3. EL ESTADO DEL ARTE

3.1. Metodología

En primer lugar, a la hora de decidir qué metodología se va a utilizar para desarrollar el proyecto hay que tener en cuenta dos ramas principales. Se puede optar por seleccionar una metodología tradicional, o una metodología ágil. La principal diferencia entre ambas suele ser la magnitud del proyecto a desarrollar.

La metodología tradicional usa un enfoque predictivo, en el que se busca que el desarrollo del proyecto se haga de manera secuencial y no haya que volver atrás para realizar los cambios. Se realizan muchas estimaciones y análisis previos, ya que es la parte más fundamental de este tipo de metodología. Son metodologías fáciles de implementar cuando se va a desarrollar algún proyecto pequeño.

La metodología ágil surge como contrapunto a la tradicional, como su propio nombre indica esta metodología se caracteriza por ser adaptativa y flexible, por lo que no es nada reticente a realizar cambios. Son metodologías más complejas de implementar pero que dan mejor resultado a la hora de desarrollar proyectos de gran magnitud, por la su característica principal, la reversibilidad.

3.1.1. Modelo en Cascada

Es un enfoque que como su propio nombre indica, ordena las etapas del desarrollo del proyecto.

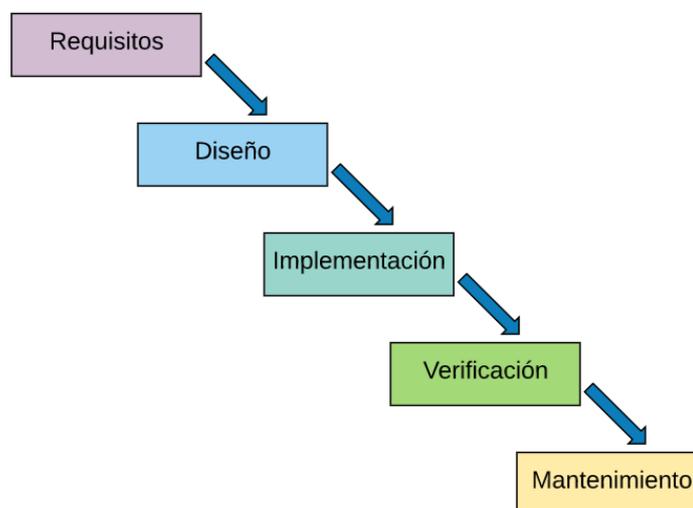


Ilustración 2. Modelo en cascada

3.1.1.1. Fases del modelo

Análisis de requisitos del software

En esta fase se analizarán las necesidades de los clientes objetivo que utilizarán el software, por lo cual se podrá conocer los puntos que se deben cubrir. En esta fase se obtendrá una memoria, dicha memoria, no entra en detalles internos, sino que es un resumen de lo que el sistema debe conseguir.

En esta etapa, se va a realizar un consenso completo de todos los requerimientos del sistema. Es una etapa muy importante, ya que es la base del proyecto, por lo que las siguientes etapas dependen de esta fase.

Diseño del sistema

En esta etapa se realiza una organización y estructuración del sistema, para que se pueda realizar por separado mediante el desarrollo en equipo. El resultado es un informe en el que se encuentra la descripción de la estructura relacional global del sistema, una especificación detallada de lo que debe realizar cada parte y su posterior combinación.

Diseño del programa

En esta fase se realizan los algoritmos que se necesitarán para cumplir los requerimientos previamente establecidos y analizar qué herramientas se van a usar en la siguiente etapa.

Codificación

En esta fase se realiza la implementación del código fuente. Se realizarán prototipos para las pruebas y corregir los errores que vayan surgiendo.

Pruebas

En esta fase se realiza el ensamble de todos los elementos programados, para la composición del sistema. Se probará todo al detalle para corregir los errores que vayan surgiendo antes de entregarlo al cliente final.

Validación y Verificación del producto de software

En esta fase el cliente prueba el sistema y comprueba que el sistema funciona correctamente.

3.1.2. Modelo de prototipos

Este modelo se suele realizar cuando el cliente no tiene una visión clara de lo que quiere, por lo que se realiza un desarrollo rápido del sistema a nivel visual, se le muestra al cliente, y así puede detectar rápidamente los errores que se han cometido y solventarlos. Como su propio nombre indica, podemos decir que este modelo se basa en la creación de un prototipo.

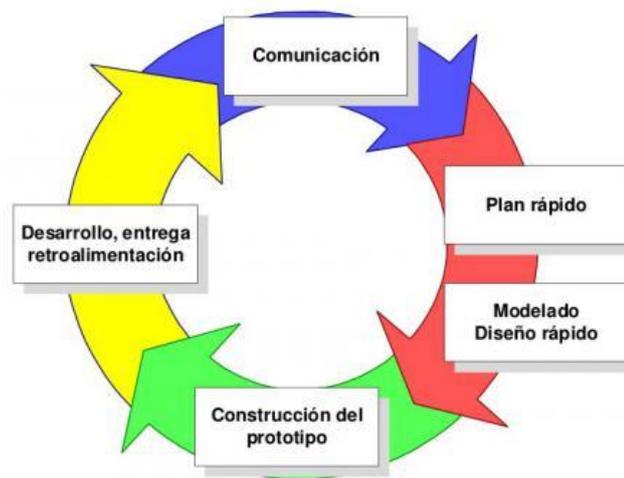


Ilustración 3. Modelo de prototipos

3.1.2.1. Etapas

- Comunicación con el cliente.
- Plan rápido.
- Modelado, diseño rápido.
- Construcción del prototipo.
- Desarrollo, entrega y retroalimentación.
- Entrega del desarrollo final.

3.1.3. Modelo en espiral

Este modelo consiste en una sucesión de ciclos que se van repitiendo formando una espiral. La interpretación usual es que en cada ciclo de la espiral se realiza un

modelo en cascada. Este modelo tiene muy en cuenta los riesgos a la hora del desarrollo de software.

Se comienza analizando las distintas alternativas de desarrollo, y se escoge la que tiene un menor riesgo, o el riesgo es más fácilmente asumible. Así se realizaría el primer ciclo de la espiral. Cuando el cliente sigue avanzando, volvemos al paso anterior en el que volvemos a analizar las nuevas alternativas y riesgos. Este proceso se repite hasta que el proyecto se haya desarrollado y no sea necesaria la realización de más ciclos.

En cada ciclo se realizarán cuatro tareas:

- Determinar objetivos
- Análisis del riesgo
- Desarrollar y probar
- Planificación

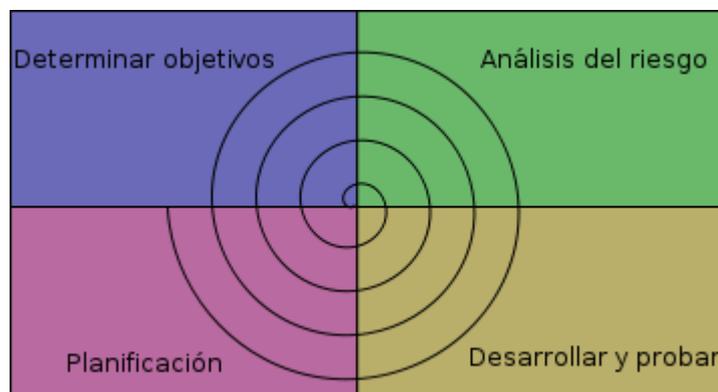


Ilustración 4. Modelo en espiral

3.1.4. Modelo Incremental

Este modelo surgió como respuesta a los fallos que presentaba el modelo en cascada.

La planificación del proyecto, se realiza en bloques temporales llamados iteraciones. En cada iteración se repite un proceso de trabajo que brinda un resultado con un acabado mejor de cara al producto final.

Para que se consiga esto, cada requerimiento se desarrollará por completo en una única iteración, la cual debe incluir pruebas y una documentación para que el equipo de desarrollo pueda lograr los objetivos requeridos, y se le pueda entregar al cliente.

En cada iteración se realizarán cambios en el diseño, y se agregarán nuevas funcionalidades y capacidades al sistema, logrando así un mejoramiento mucho más completo.

Este modelo se basa en dos premisas:

- Los usuarios finales no saben lo que buscan para satisfacer sus necesidades.
- Durante la etapa de desarrollo, los procesos pueden cambiar.

El modelo consta de tres etapas:

- Etapa de inicialización.
- Etapa de iteración.
- Lista de control de proyecto.

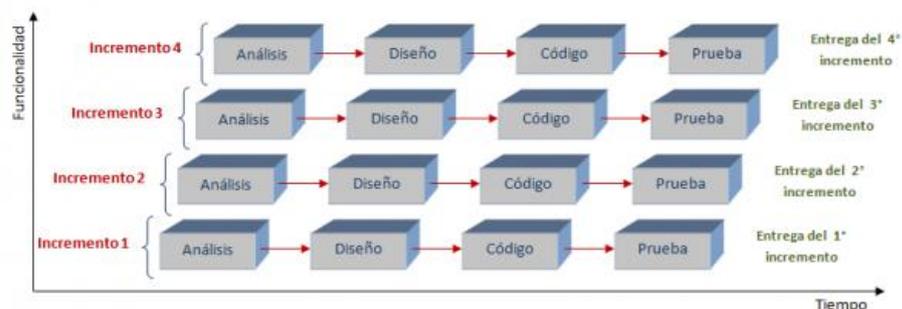


Figura 1: El Modelo Incremental

Ilustración 5. Modelo incremental

3.2. Arquitectura

Se realizará la comparativa de diferentes frameworks que se podrían utilizar para realizar la aplicación web. Con los resultados de esta comparación se elegirá el que más se adecúe a nuestro proyecto.

Para la implementación del proyecto se debe conocer el patrón MVC (Modelo-Vista-Controlador), el cual es un patrón de arquitectura software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello, este patrón propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información y por otro lado para la interacción del usuario. Este patrón se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. La principal característica de este patrón es la separación que realiza de los datos y la lógica de negocio de su representación y el módulo encargado de gestionar los eventos y las comunicaciones.

A continuación, se presentan los componentes modelo, vista y controlador en detalle:

- El Modelo: gestiona los accesos a la información con la cual el sistema opera (consultas y actualizaciones). Implementa los privilegios de acceso que se describan en la lógica de negocio. Es el encargado de enviar a la vista la información solicitada para ser mostrada. Las peticiones de acceso llegan al modelo a través del controlador.
- El Controlador: da respuesta a eventos, normalmente son acciones de usuario, y llama al modelo cuando se realiza una solicitud sobre la información. También envía comandos a la vista que tenga asociada, si se ha solicitado un cambio en la forma en que se representa el modelo. Con esta definición se podría decir que el controlador hace de intermediario entre la vista y el modelo.
- La Vista: realiza la presentación de la información y lógica de negocio (modelo) de forma que sea posible la interacción mediante una interfaz

de usuario, por tanto, requiere la información del modelo que debe representar como salida.

Flujo de control que se sigue en este patrón:

- a. El usuario interactúa con la interfaz.
- b. El controlador recibe una propuesta de acción solicitada por el usuario. Dicho controlador, realiza una gestión del evento que llega a través de un callback o un gestor de eventos.
- c. El controlador realiza un acceso al modelo, el cual lo actualiza o modifica según la acción que haya realizado el usuario.
- d. Los objetos de la vista despliegan la interfaz de usuario. Dicha vista recibe los datos proporcionados por el modelo, los cuales generan una interfaz apropiada para el usuario, donde se muestran los cambios realizados en el modelo.
- e. La interfaz se mantiene en espera de que el usuario realice nuevas interacciones, volviendo así al primer punto.

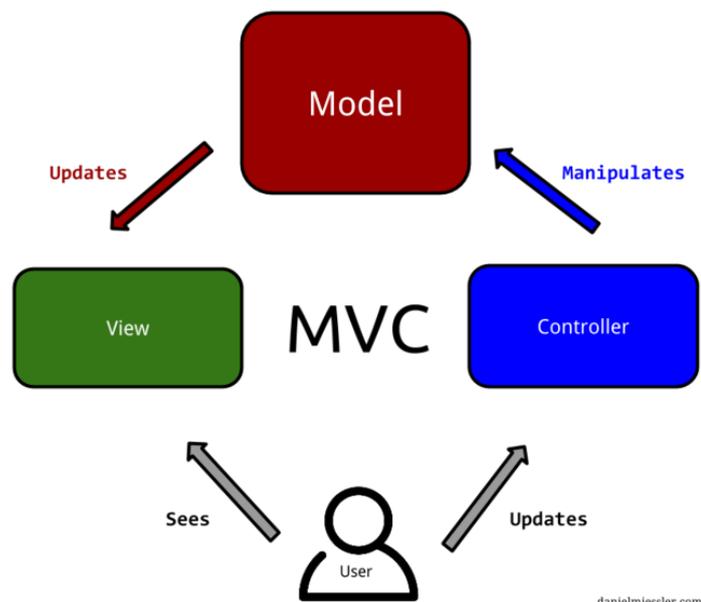


Ilustración 6. Flujo de control patrón MVC

El hecho de escoger un patrón MVC y que sea una aplicación web, viene motivado por la necesidad de desarrollar una aplicación que sea operativa desde cualquier dispositivo, para ello, la aplicación se ha desarrollado de manera responsiva sin un coste desorbitado.

3.3. Frameworks

3.3.1. Codeigniter



Ilustración 7. Logo de Codeigniter

Codeigniter es un framework PHP que usa la arquitectura MVC. Permite construir aplicaciones web altamente escalables con un uso menor del espacio.

Codeigniter es el framework utilizado por su sencillez tanto a la hora de su aprendizaje como de la rapidez de inicio. Es uno de los frameworks con más documentación detallada, además de ser utilizado por una gran comunidad de desarrolladores. Ofrece un rendimiento sólido, por lo que es perfecto para construir aplicaciones ligeras.

Características principales:

- Es un framework ligero.
- Sencillo y rápido de iniciar, gracias a su simplicidad y a la excelente documentación.
- Se crean aplicaciones escalables usando la arquitectura MVC.

3.3.2. Zend



Ilustración 8. Logo de Zend

Es un framework orientado a objetos, y basado en MVC que permite cargar solamente los componentes que se necesitan como librerías individuales.

Gracias a esto, y a su naturaleza orientada a objetos, se puede hacer una reutilización de mucho código que hayas escrito. Además tiene una manera muy sencilla de integrar librerías externas.

Principales características:

- Es un framework orientado a objetos con arquitectura MVC.
- Se puede reutilizar código debido a su diseño.
- Se pueden integrar librerías externas con facilidad.
- Utilización de los componentes que se necesitan en cada momento.

3.3.3. Phalcon



Ilustración 9. Logo de Phalcon

Es uno de los frameworks más rápidos que existen, debido a que su código fuente está escrito en C, por lo que es una extensión en C de PHP.

Consume pocos recursos, y también usa una arquitectura MVC. Al realizarse la instalación, solamente se instalan los archivos básicos. El resto de módulos se van añadiendo según se necesiten en cada momento.

Características principales:

- Framework PHP basado en C.
- Necesita pocos recursos.
- Solo utiliza los módulos y librerías que necesita en cada momento.

3.3.4. CakePHP



Ilustración 10. Logo de CakePHP

Es el primer framework PHP con arquitectura MVC que salió al mercado.

CakePHP ofrece una gran cantidad de librerías que incluyen muchos componentes útiles. (Terminar)

Principales características:

- Gran cantidad de componentes.

3.4. Elección de Codeigniter

Tras realizar la comparativa entre los distintos frameworks PHP con arquitectura MVC existen, Codeigniter ha sido la elección principal.

El motivo de dicha elección es la sencillez de uso, la gran comunidad existente de desarrolladores que lo utilizan y lo fácil que es el aprendizaje, y el consiguiente desarrollo de aplicaciones web.

4. TECNOLOGÍAS UTILIZADAS

4.1. PHP



Ilustración 11. Logo de PHP

PHP ha sido el lenguaje de programación utilizado por excelencia para el proyecto, en su versión 7.1. Es un lenguaje de código abierto adecuado para el desarrollo de aplicaciones web dinámicas, y puede ser incrustado en HTML.

PHP es ejecutado en el lado del servidor, por lo que genera el HTML y se lo envía al cliente por tanto el cliente recibe el resultado del script. El código es interpretado por un servidor web que genera el texto plano en formato UTF-8. Dicho código PHP es invisible al navegador web y al cliente.

Es un lenguaje simple de utilizar y de aprender, pero a su vez ofrece un amplio rango de características más avanzadas para los desarrolladores profesionales.

PHP está pensado para emplearse en todos los SO. También admite la mayoría de servidores web que existen.

Una de las características más reseñables es el soporte del que dispone para una gran cantidad de bases de datos.

4.2. JavaScript



Ilustración 12. Logo de JavaScript

Este lenguaje se ha usado para realizar las funciones en las vistas que nos permiten de forma dinámica, visualizar los datos que se van introduciendo.

Es un lenguaje interpretado de alto nivel, de acuerdo con la especificación ECMAScript. JavaScript se define como un lenguaje orientado a objetos, basado en prototipos, débilmente tipado y dinámico.

JavaScript tiene APIs para trabajar con texto, arrays, fechas, expresiones regulares y el DOM.

Inicialmente y principalmente, se usa en su forma del lado del cliente, implementado como parte de un navegador web, aunque actualmente ahora los motores JavaScript están embebidos en otros tipos de software, incluyendo el lado del servidor en servidores web y bases de datos, y en programas no web como procesadores de texto y software PDF.

4.3. HTML5



Ilustración 13. Logo de HTML5

HTML5 se ha utilizado para la creación de las vistas de la aplicación.

Es el lenguaje de marcado para la elaboración de páginas web. Se puede complementar con tecnologías como CSS y lenguajes de scripts como JavaScript.

Los navegadores reciben los documentos HTML de parte de un servidor web o de manera local y renderizan los documentos como páginas web multimedia. HTML describe la estructura de una web de forma semántica.

HTML5 es la quinta y actual versión de HTML. Incluye modelos de procesamiento detallados, para desarrollar implementaciones más interoperables, extiende, mejora y racionaliza el marcado disponible para los documentos, e introduce marcado e interfaces de programación de aplicaciones (APIs) para las aplicaciones web más complejas.

4.4. CSS



Ilustración 14. Logo de CSS

Es un lenguaje de hoja de estilo usado para describir la presentación de un documento escrito en un lenguaje de marcado como HTML.

CSS se ha diseñado para permitir la separación entre presentación y contenido, incluyendo layout, colores y fuentes. Esta separación mejora la accesibilidad del contenido, proporciona más flexibilidad y control en la especificación de las características de presentación, permite que múltiples páginas web compartan formato, especificando el CSS relevante en un fichero .css separado, y reduce la complejidad y la repetición en el contenido estructural.

Se ha utilizado junto al código HTML para proporcionar a las vistas de estilo.

4.5. Bootstrap



Ilustración 15. Logo de Bootstrap

Es un framework CSS gratuito y open-source, responsivo, mobile-first para desarrollo web front-end. Contiene CSS y plantillas basadas en JavaScript para tipografías, botones, formularios, navegación y otros componentes de interfaz.

En este proyecto, se ha utilizado la versión 4 de Bootstrap, en concreto se han utilizado las plantillas de ADMIN LTE2, para la creación de los formularios, tablas, y el resto de elementos visuales que se encuentran en la aplicación, tales como los modales o las alertas.

4.6. XAMPP



Ilustración 16. Logo de XAMPP

Es un paquete gratuito de software libre desarrollado por Apache Friends, el cual consiste principal mente en el servidor Apache HTTP, una base de datos MariaDB, e intérpretes de scripts escritos en los lenguajes de programación PHP y Perl. La mayoría de servidores webs actuales usan los mismos componentes que XAMPP, por lo que hace la transición desde un servidor local a un servidor web posible.

Se ha utilizado para crear un servidor local, donde poder lanzar la plataforma, también se ha usado la base de datos que proporciona la aplicación para crear la que se ha usado en el proyecto.

4.7. Codeigniter

Es un framework web de software libre para la construcción de aplicaciones web dinámicas con PHP basado en el patrón MVC.

Este framework se ha utilizado para desarrollar la web de una manera sencilla, ya que es muy intuitivo y de fácil aprendizaje.

4.8. Visual Paradigm



Ilustración 17. Logo de Visual Paradigm

Es una herramienta de UML CASE que incluye MUL 2, SysML y (BPMN) de Object Management Group (OMG). Aparte de poder realizar modelado, también proporciona generación de informes, e incluye la generación de código.

Dicha herramienta ha sido utilizada para la creación de la planificación, análisis y diseño del proyecto.

4.9. GanttProject



Ilustración 18. Logo de GanttProject

Es un programa de código abierto con licencia GPL escrito en Java, cuyo objetivo es la administración de proyectos usando el diagrama de Gantt.

4.10. Conclusión acerca de las tecnologías utilizadas

La selección de las tecnologías mencionadas anteriormente, ha sido fruto de una búsqueda exhaustiva para determinar cuales eran las más adecuadas para la realización tanto de la aplicación como del proyecto. En el siguiente capítulo, se detalla el funcionamiento de las tecnologías usadas para el desarrollo de la aplicación.

5. DESARROLLO

En esta parte de la memoria se va a dar comienzo al desarrollo del proyecto a partir del modelo incremental, el cual como se ha descrito antes, está compuesto por una serie de incrementos con cuatro fases en cada incremento. Análisis, en el cual se determinarán los requisitos que va a necesitar el incremento. Diseño, en el cual se definirá la estructura de la solución, a raíz de los requisitos propuestos en la fase de análisis, aparte, en esta fase se desarrollará la arquitectura de dicha solución. Codificación, es la fase en la que se va a implementar el código fuente de la aplicación, siguiendo el diseño previamente realizado. Pruebas, es la fase en la cual, como su propio nombre indica, se realizarán una serie de pruebas para analizar si el resultado obtenido es el que se buscaba inicialmente.

5.1. Primer incremento

A menudo el primer incremento suele ser uno de los más importantes del proyecto, dado que se asentarán las bases del proyecto a partir de las cuales se irá trabajando en el resto de incrementos. Por ello es necesario definir los requisitos, tanto funcionales como no funcionales que debe satisfacer nuestro prototipo. En esta etapa también se va a realizar un historial con las tareas que se van a realizar, y unos objetivos definidos.

5.1.1. Análisis

5.1.1.1. Requisitos Funcionales

Los requisitos funcionales están definidos como aquellos que reseñan las actividades que el proyecto debe efectuar, es decir, la funcionalidad que realizará el proyecto al efectuarse determinadas circunstancias.

Los requisitos que se han podido observar, en base a otros ejemplos, y a el estudio realizado sobre las necesidades de los centros educativos son los siguientes:

- a. El administrador es el encargado de crear los usuarios al principio del curso escolar, a través de un formulario de registro. También está encargado de asociar los alumnos con sus respectivos profesores.

- b. El sistema resultante debe ser accesible a través de cualquier tipo de dispositivo.
- c. Solamente los usuarios registrados por el administrador tendrán acceso al sistema, cada uno a una vista dependiendo del tipo de usuario que sea (profesor, alumno, o tutor legal).
- d. La funcionalidad de cada usuario varía según el rol de usuario.
 - a. Si el usuario es un profesor, puede crear los logros, asignarlos y ver un listado de los alumnos asociados a él. Aparte tiene la función de asignar a cada alumno un tutor legal.
 - b. Si el usuario es un alumno, puede ver las medallas positivas que tiene asignadas.
 - c. Si el usuario es un tutor legal, puede ver la totalidad de medallas que tienen sus respectivos niños asociados, pueden ver las medallas positivas o negativas.

5.1.1.2. Requisitos no funcionales

Los requisitos no funcionales están definidos como los requisitos que no describen información a guardar, ni funciones a realizar, solamente características de funcionamiento.

Dentro de los requisitos no funcionales, se tienen en cuenta tales características como la eficiencia, la seguridad, la dependencia y la usabilidad del sistema.

- Requisitos no funcionales de la aplicación:
 - Usabilidad:
 - Cada tipo de usuario debe poder familiarizarse con la aplicación de una manera rápida y sencilla y así poder utilizar sus funciones preasignadas
 - Eficiencia:
 - La aplicación debe tener un tiempo de respuesta óptimo tanto en el cambio de funciones como en las consultas realizadas a la base de datos.

- Seguridad:
 - El acceso debe estar controlado por el tipo de usuario que accede a la aplicación. También debe controlar si un usuario externo desea acceder a alguna vista sin estar registrado por el administrador. Las contraseñas se cifran cuando un usuario es dado de alta.
 - El sistema debe estar disponible las 24 horas del día.

5.1.1.3. Diagrama de casos de usos

Se van a realizar en cada incremento diagramas de casos de uso para la ilustración de las actividades que van a desarrollar los distintos usuarios para llevar a cabo un proceso. Estos diagramas son utilizados para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los otros usuarios y/o con otros sistemas.

Puesto que se trata de la etapa de inicialización, el diagrama presentado a continuación, será un diagrama base, que muestra desde una perspectiva global, las actividades que se realizarán a lo largo de los distintos incrementos, teniendo en cuenta los requisitos previamente citados. En cada incremento futuro, se realizará el análisis, diseño, implementación y prueba de las actividades que se muestran a continuación.

Cada tipo de usuario, está relacionado con una o varias actividades, dependiendo de los permisos que este reciba, como se muestra en la siguiente ilustración:

Visual Paradigm Standard (Antonio (Universidad de Jaén))

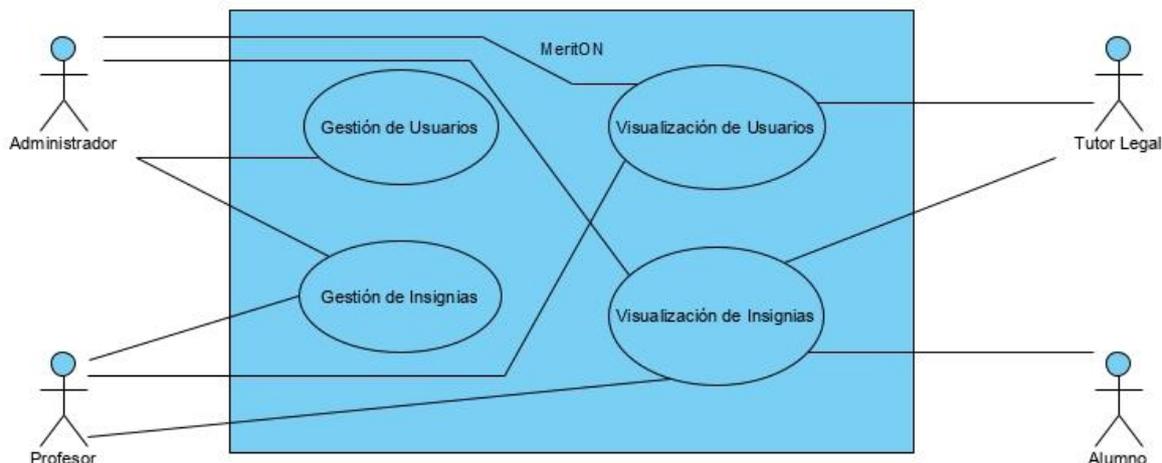


Ilustración 19. Diagrama de casos de uso inicial

5.1.2. Diseño

En la etapa de inicialización se están sentando las bases del proyecto, por lo que se va a proceder a la realización de una base de datos que sea capaz de almacenar y relacionar la información sobre los alumnos, los padres, los tutores y las insignias que se van a ir asignando a cada alumno. En este apartado de diseño también se realizara el diseño de la interfaz de usuario que se deberá de implementar en incrementos posteriores.

5.1.2.1. Base de datos

Se va a proceder a exponer las especificaciones de cada una de las tablas creadas.

- Tabla de usuario: contiene la información acerca de los usuarios.
- Tabla de insignias: contiene la información acerca de las insignias.
- Tabla de datos del alumno: contiene la relación entre los alumnos y las insignias asignadas a ellos.
- Tabla de profesor y alumno: contiene la relación entre los alumnos y los profesores.
- Tabla de tutores legales y alumnos: contiene la relación entre los alumnos y sus tutores legales.

Tabla de usuario

La tabla cuyas especificaciones se presentan a continuación consta de los datos acerca de los usuarios.

- id_usuario: identificador único y clave primaria.
- nombre: nombre del usuario.
- apellidos: apellidos del usuario.
- f_nacimiento: fecha de nacimiento del usuario.
- dni: DNI del usuario sin letra.
- rol: rol que va a desempeñar el usuario. Puede ser de tres tipos:
 - 0 - Profesor: puede crear Insignias, modificarlas, asignarlas, asignar a sus alumnos con sus tutores legales.
 - 1 - Padres: puede ver las insignias positivas y negativas de sus hijos.
 - 2 - Alumnos: pueden ver solo y exclusivamente sus insignias positivas.
- alias: alias del usuario.
- contraseña: contraseña para acceder del usuario.

Tabla de insignias

La tabla cuyas especificaciones se presentan a continuación consta de los datos acerca de las insignias.

- id_insignia: identificador único y clave primaria.
- nombre: nombre de la insignia.
- descripcion: descripción de la insignia.
- tipo: la insignia puede tener dos valores distintos. Por defecto será positivo.
 - 0: la insignia tiene valor positivo.
 - 1: la insignia tiene un valor negativo.
- ruta: es la ruta en la que se guarda el icono de la insignia.

Tabla de datos alumno

La tabla cuyas especificaciones se presentan a continuación consta de la relación que tienen los alumnos con las insignias y la fecha en la que ha sido asignada cada insignia. Los datos que contienen son:

- id del alumno
- id de la insignia
- fecha en la que ha sido asignada.

Tabla de profesor y alumno

La tabla cuyas especificaciones se presentan a continuación consta de la relación que tienen los alumnos y los profesores. Está compuesta por:

- id del profesor
- id del alumno.

Tabla de tutores legales y alumnos

La tabla cuyas especificaciones se presentan a continuación consta de la relación que tienen los alumnos y sus tutores legales. Está compuesta por:

- id del tutor legal
- id del alumno.

El esquema relacional de la base de datos corresponde con el expuesto en la Ilustración 30

5.1.2.2. Diagrama de Secuencia

Un diagrama de secuencia muestra la interacción de los objetos en una aplicación a través del tiempo y se modela para cada caso de uso. En este incremento, dado que es la etapa de inicialización, se va a realizar el diagrama de Secuencia para el caso inicial, que sería el acceso a la aplicación con un usuario.

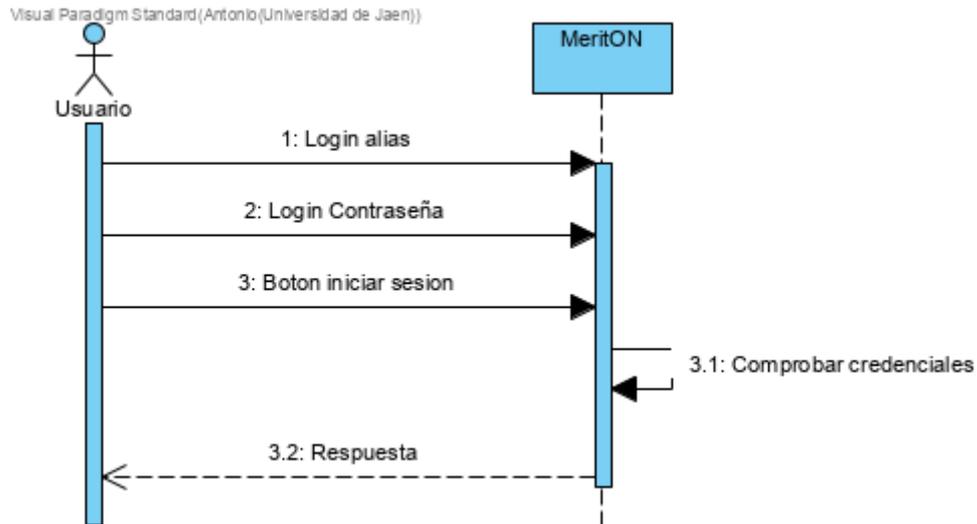


Ilustración 20. Diagrama de secuencia inicio de sesión.

5.1.2.3. Storyboard

El storyboard se trata de un conjunto de ilustraciones mostradas en secuencia, que guían al cliente mostrándole una versión de la interfaz inicial. En dicho storyboard, el cliente puede “interactuar” y ver como se desarrollan las distintas opciones.

La Ilustración 21, corresponde a la vista de la página de inicio de sesión.

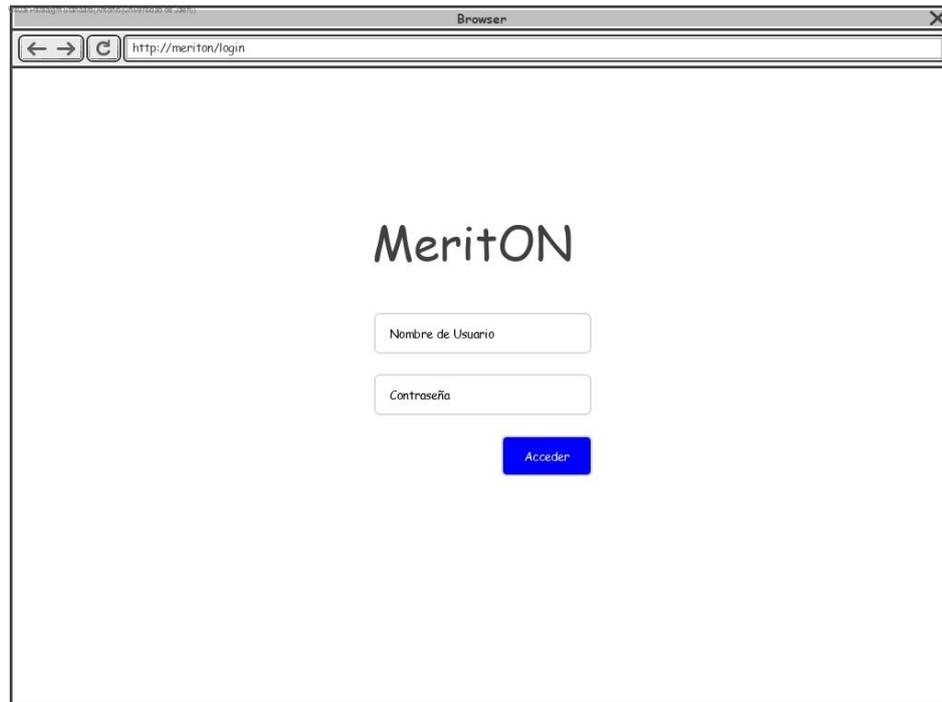


Ilustración 21. Storyboard login.

La Ilustración 22 será la vista principal del administrador, en la que se muestra el formulario de creación de usuarios.

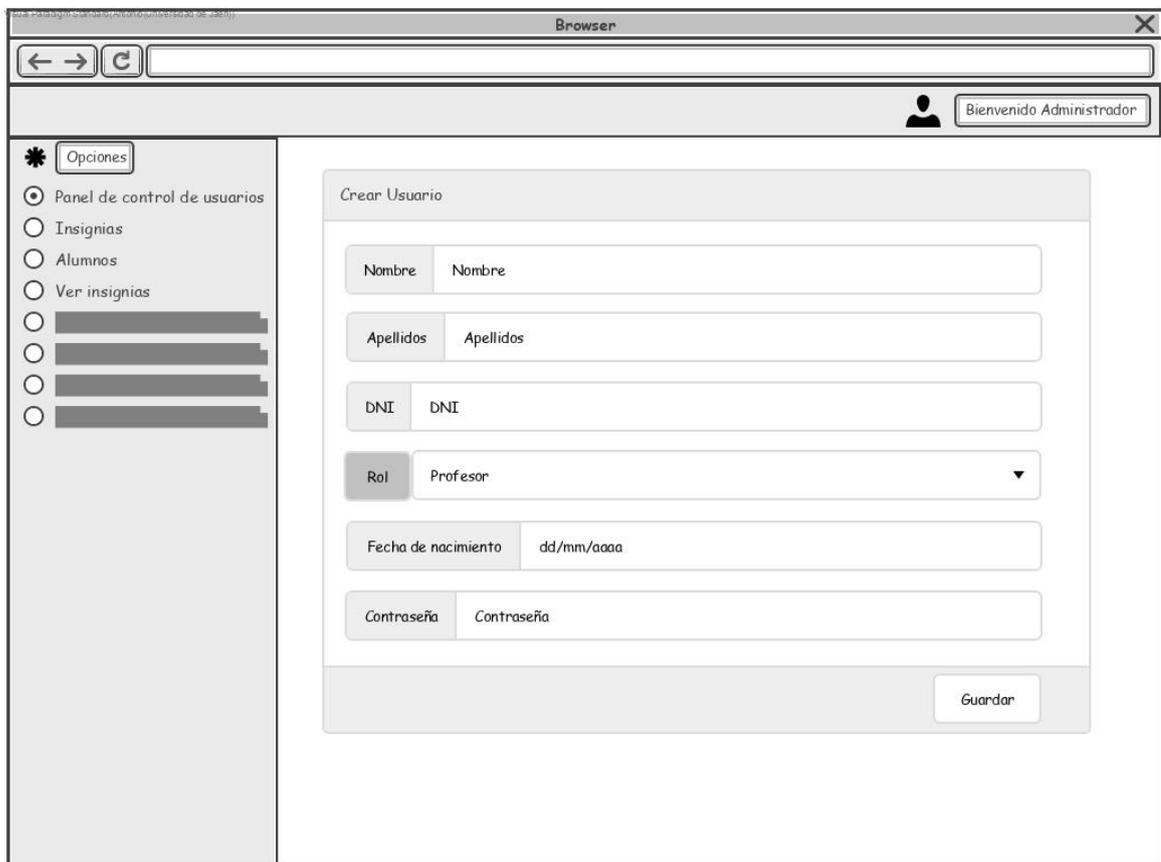


Ilustración 22. Panel de control de administrador (formulario de creación de usuarios).

La Ilustración 23 es una vista accesible solamente por parte del administrador, en esta vista se muestra una tabla con el listado de alumnos, a cada alumno se le puede asignar un profesor.

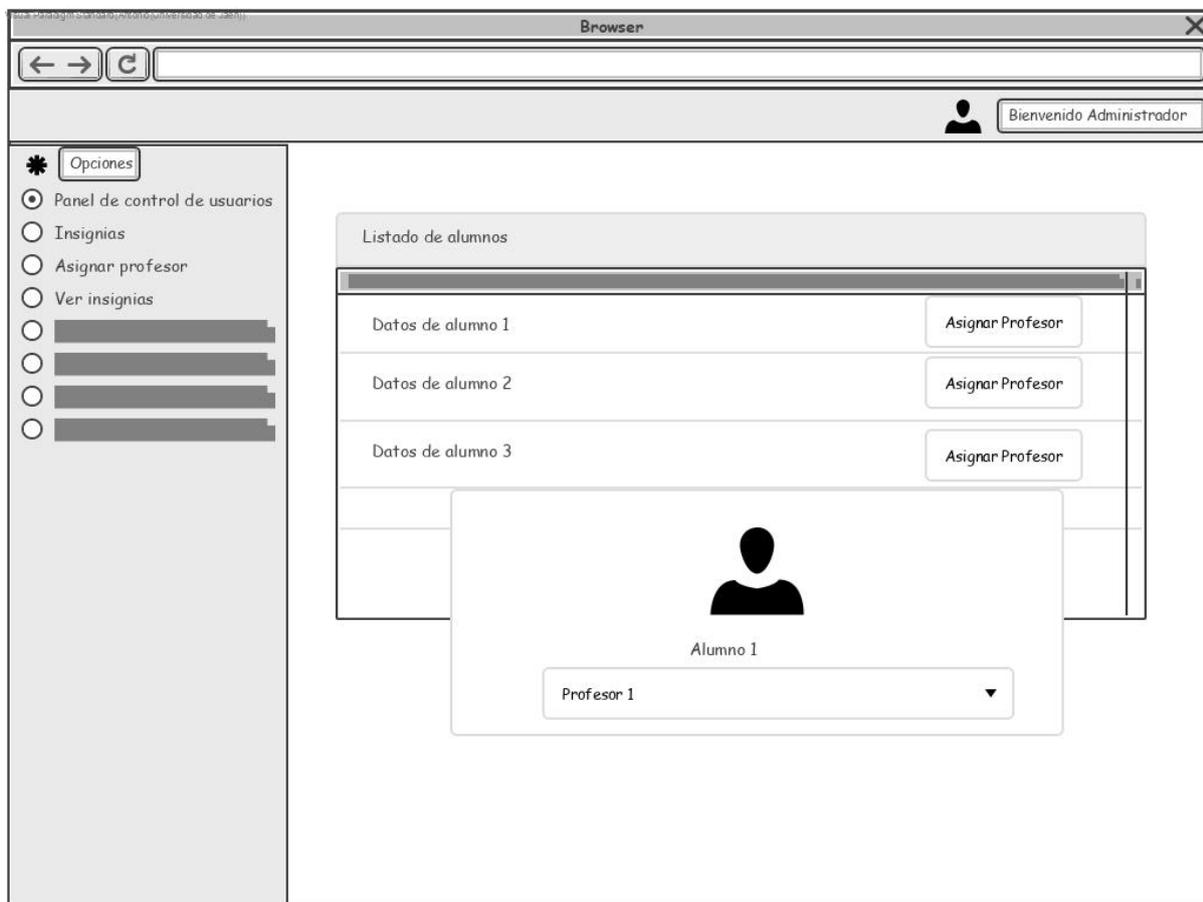


Ilustración 23. Vista de asignar profesores

Las siguientes vistas son accesibles también por el administrador, pero ya serían específicas de los distintos tipos de usuarios. En la Ilustración 24 se puede apreciar el formulario de creación de las insignias. Es exclusivo del administrador y de los profesores.

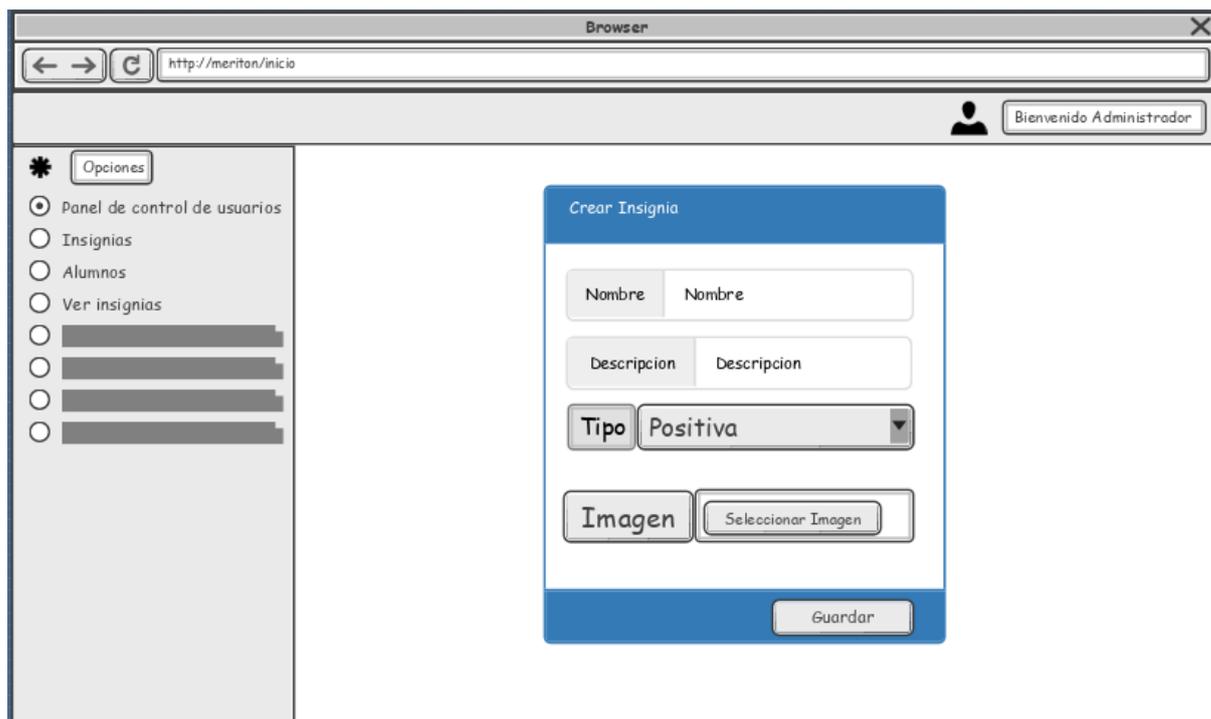


Ilustración 24. Panel de creación de Insignias

En la Ilustración 25 se muestra el panel en el que aparecen todas las insignias creadas, tanto sus imágenes como su nombre y descripción. Esta vista es accesible, tanto para el administrador como por el profesorado.

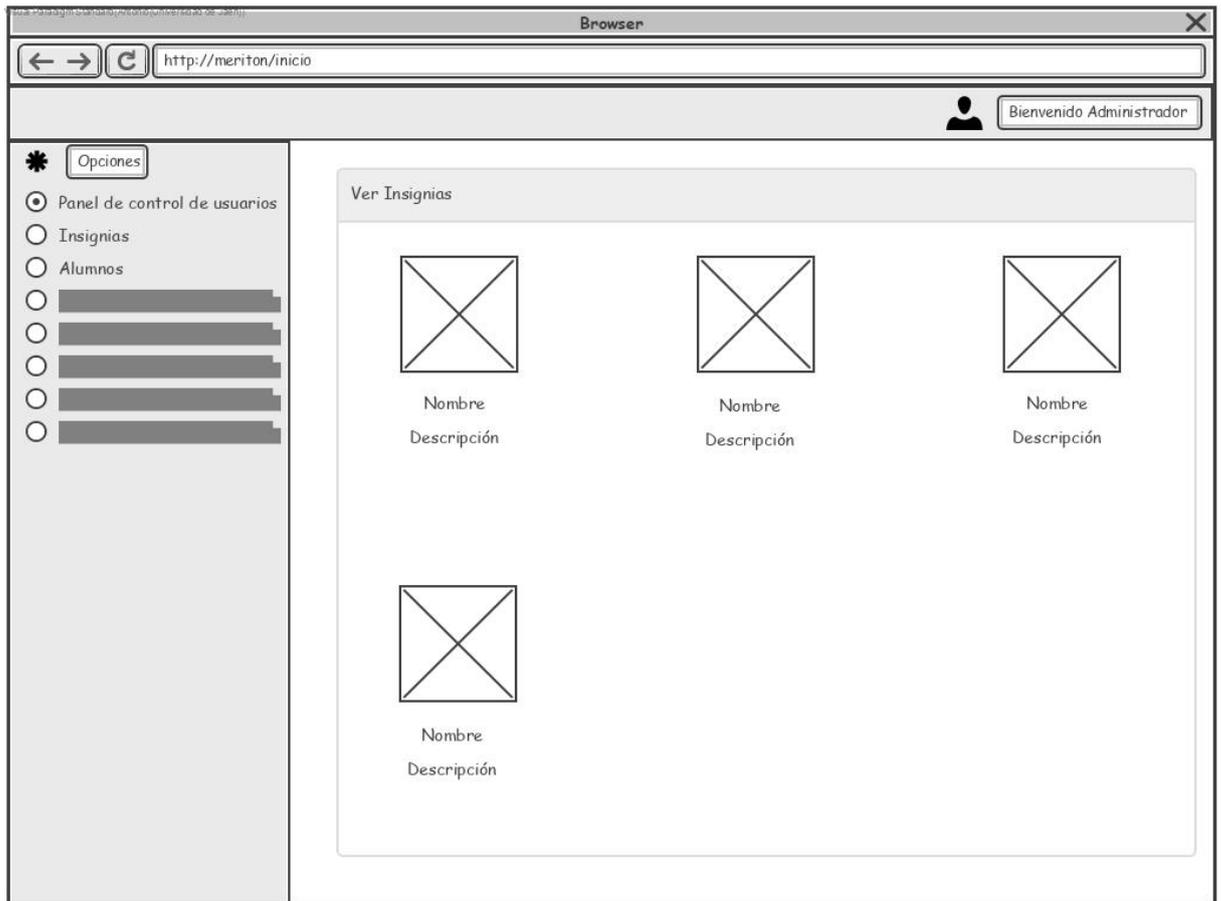


Ilustración 25. Vista de Insignias

La siguiente vista, es exclusiva de los profesores, en ella los profesores pueden asignar a cada alumno con su respectivo tutor legal.

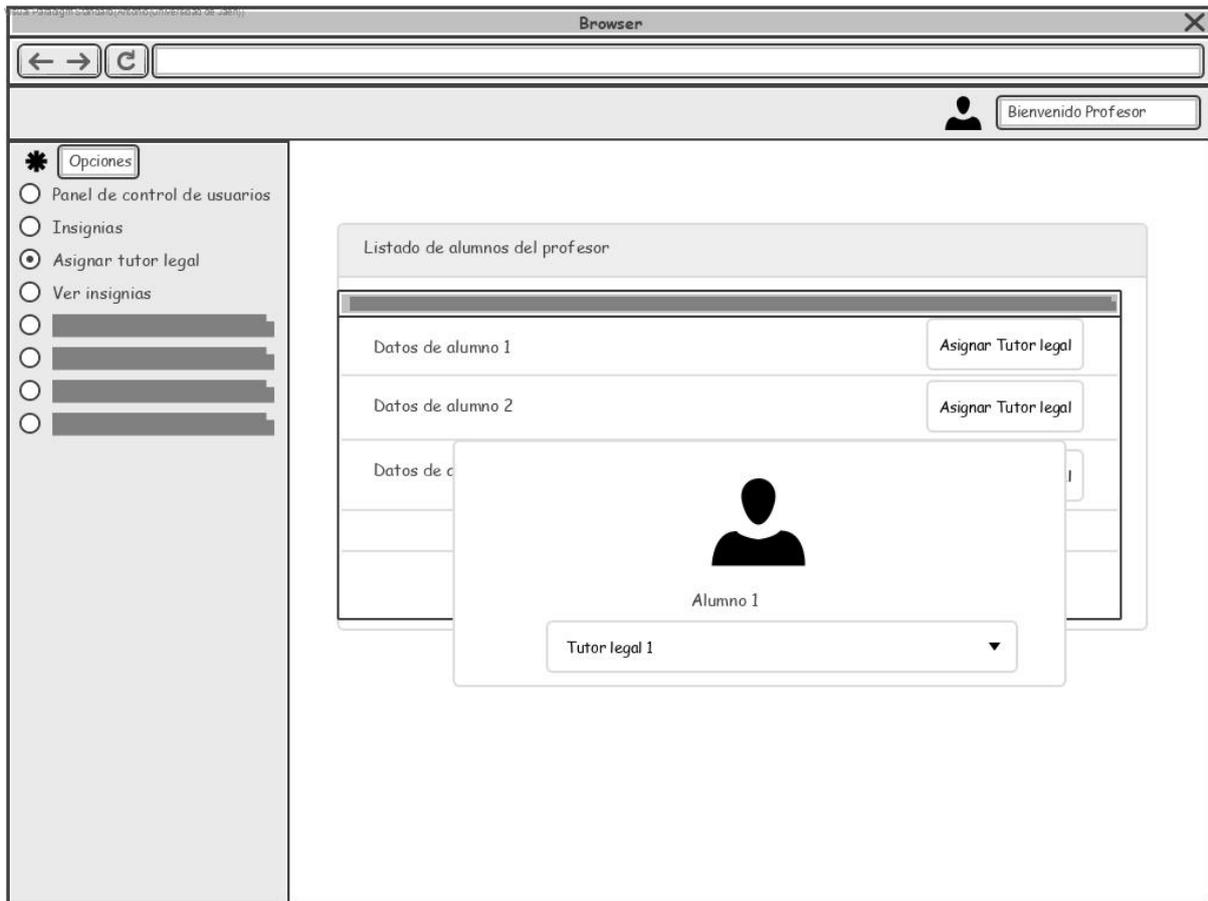


Ilustración 26. Vista de asignación de tutores legales.

La Ilustración 27 corresponde a la asignación de insignias por parte de los profesores, en esta vista aparece un listado de los alumnos de cada profesor, y en ella el profesor puede asignar las insignias que se encuentran en la base de datos a cada alumno.

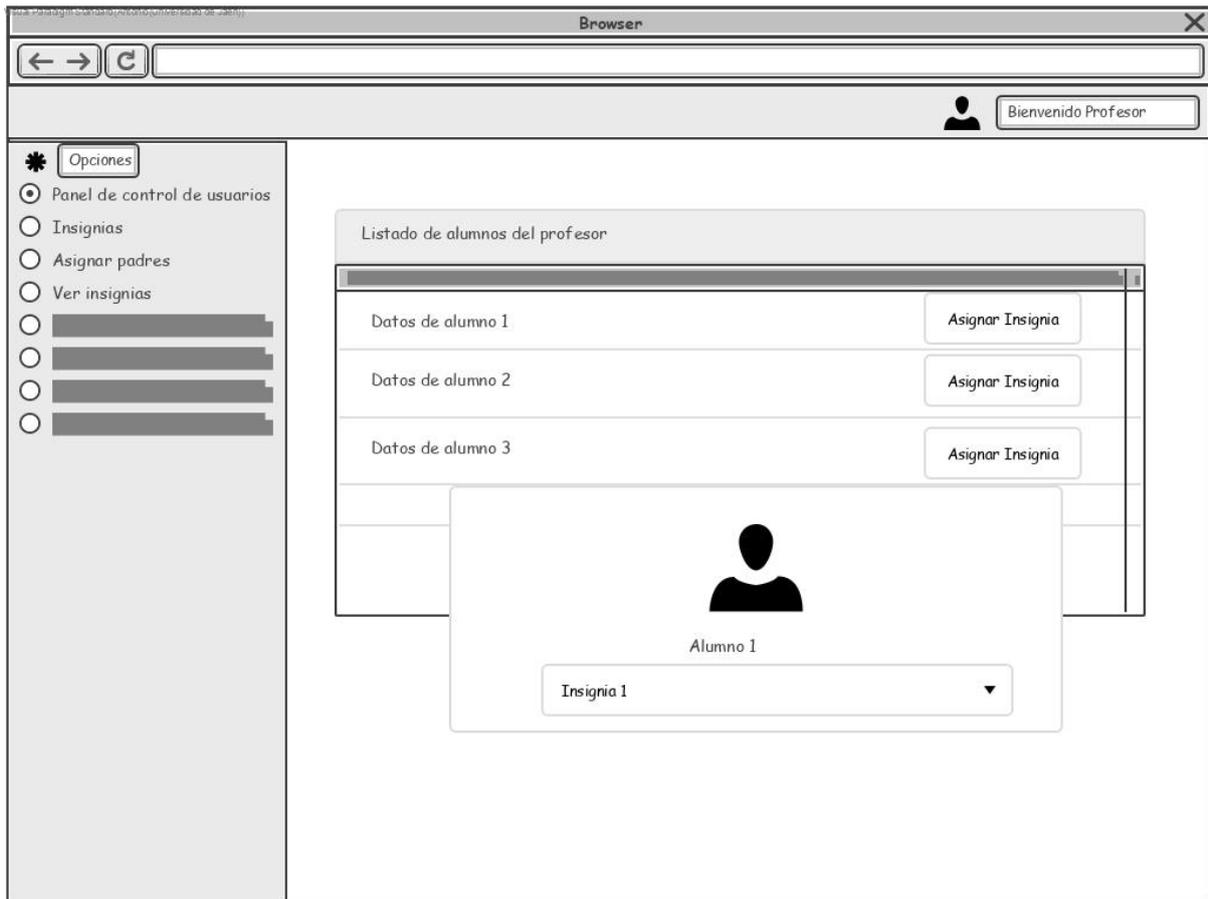


Ilustración 27. Vista de asignación de insignias

La Ilustración 28 se trata de la vista principal del alumno, en la que solamente puede ver sus insignias positivas asignadas.

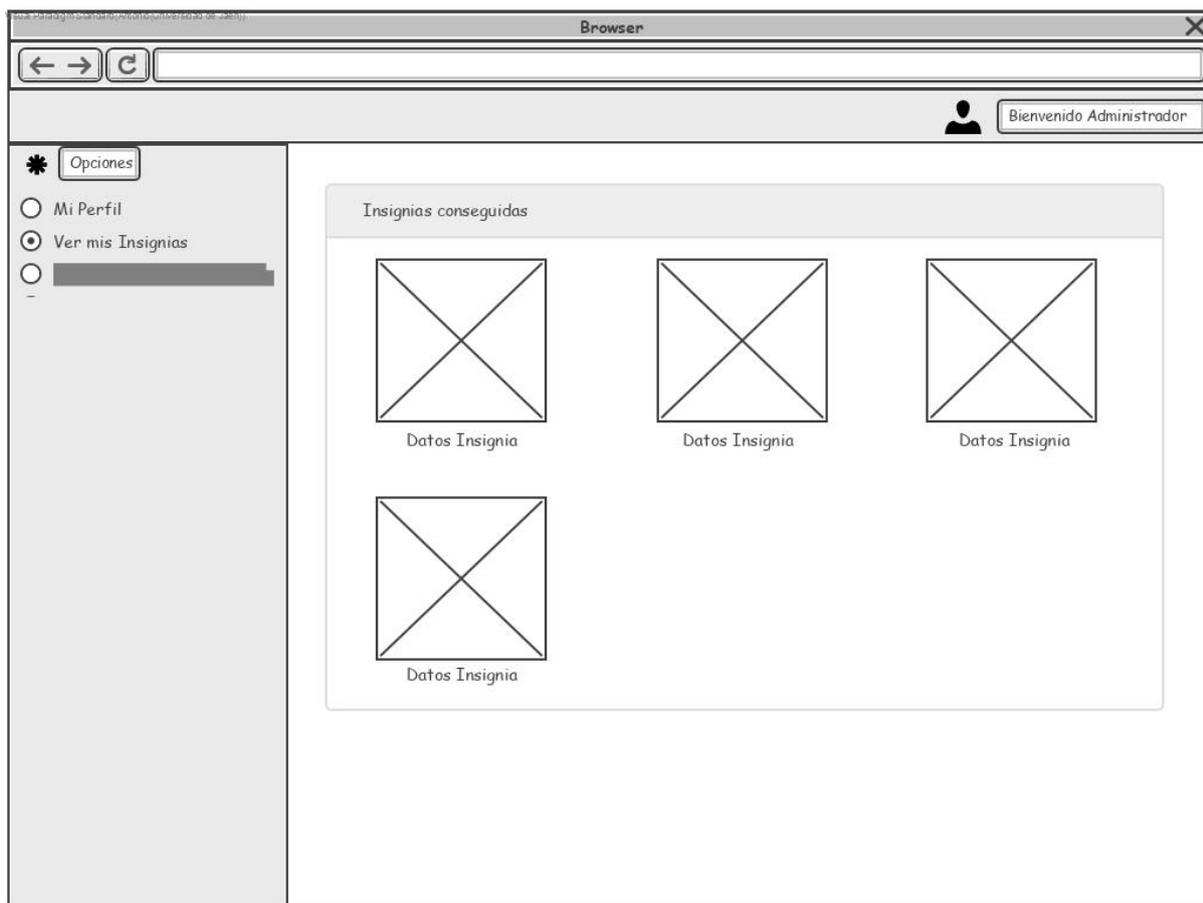


Ilustración 28. Vista principal alumno.

La Ilustración 29 es la vista principal de los tutores legales, en la cual se muestra una tabla con todos los alumnos que tiene el tutor legal correspondiente, y por cada alumno se puede acceder a todas las insignias que tiene, tanto las positivas como las negativas.

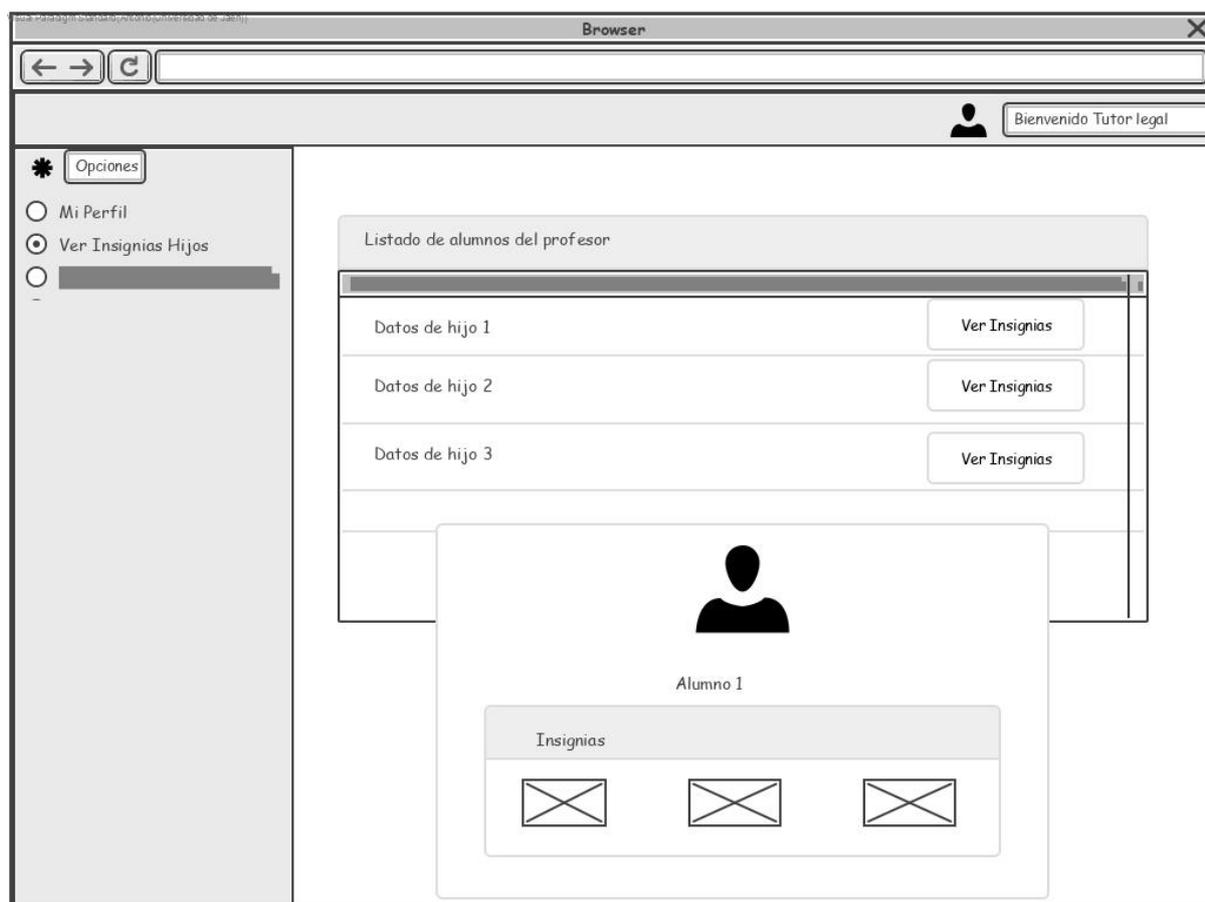


Ilustración 29. Vista principal tutor legal

5.1.3. Implementación

Al ser la primera implementación del prototipo, en lo relacionado a la base de datos, se procederá a la creación de las diferentes tablas de la base de datos, especificando sus atributos y estableciendo las relaciones existentes entre ellas. En cuanto a la página web, se procederá a la inicialización de la misma poniendo en marcha el framework codeigniter e incluyendo la plantilla seleccionada dentro del proyecto. Además, se implementará la funcionalidad de inicio de sesión especificada en el diagrama de secuencia de la fase de diseño.

5.1.3.1. Base de datos

Como podemos ver en la Ilustración 30 se ha realizado la implementación de la base de datos, de acuerdo a las tablas diseñadas anteriormente.

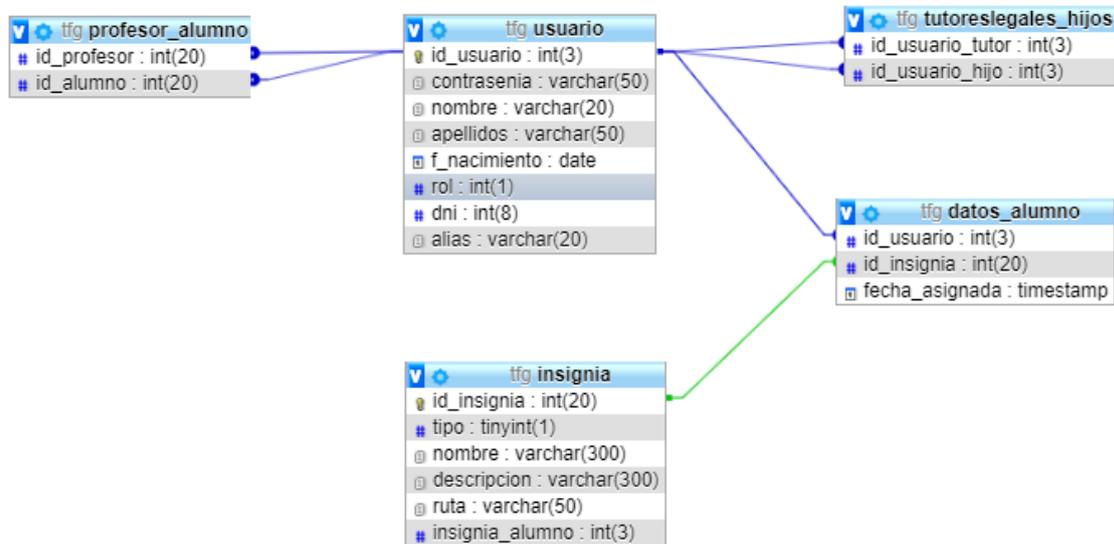


Ilustración 30. Diseño de la Base de datos

5.1.3.2. Aplicación

Atendiendo a lo comentado anteriormente en la planificación del proyecto, se ha utilizado el patrón de diseño MVC.

El primer paso realizado antes de proceder a la implementación de la aplicación es instalar el software citado anteriormente, en este caso hemos usado XAMPP, para poder desarrollar de forma local. Una vez instalado el software, se da paso a la instalación del framework utilizado, en este caso Codeigniter.

Una vez instalados los diferentes softwares, se van a crear los distintos modelos para cada tabla que contenga información en la base de datos, pero no para las tablas de relaciones.

- modeloUsuario: creada para las consultas sobre los usuarios.
- modeloInsignias: creada para las consultas sobre las insignias.

 modeloInsignias.php	02/07/2019 16:16	Archivo PHP	3 KB
 modeloUsuario.php	02/07/2019 19:49	Archivo PHP	3 KB

Ilustración 31. Modelos PHP

Para el estilo de las vistas, se ha usado una plantilla Bootstrap llamada AdminLTE2, en esta plantilla hay definidos por defecto, como en cualquier aplicación de administración web, un header, un footer y un menú, los cuales serán usados para todas las vistas y personalizados. Estas vistas serán almacenadas en una subcarpeta dentro de la carpeta views del proyecto, llamada layout.

El header está formado por una barra de navegación superior, en la cual aparecerá Bienvenido y el nombre del usuario logueado en ese momento.

En el menú se muestran las distintas opciones con las que cuenta cada usuario. El resultado es el que aparece en la Ilustración 32.



Ilustración 32. Header y menú lateral.

El siguiente paso es crear un inicio de sesión, esta vista, se crea en la carpeta views. En esta aplicación los usuarios no pueden darse de alta por sí mismos, por lo que será necesario que el administrador los haya registrado previamente en la Base de Datos. En la Ilustración 33 se muestra la vista final de login.

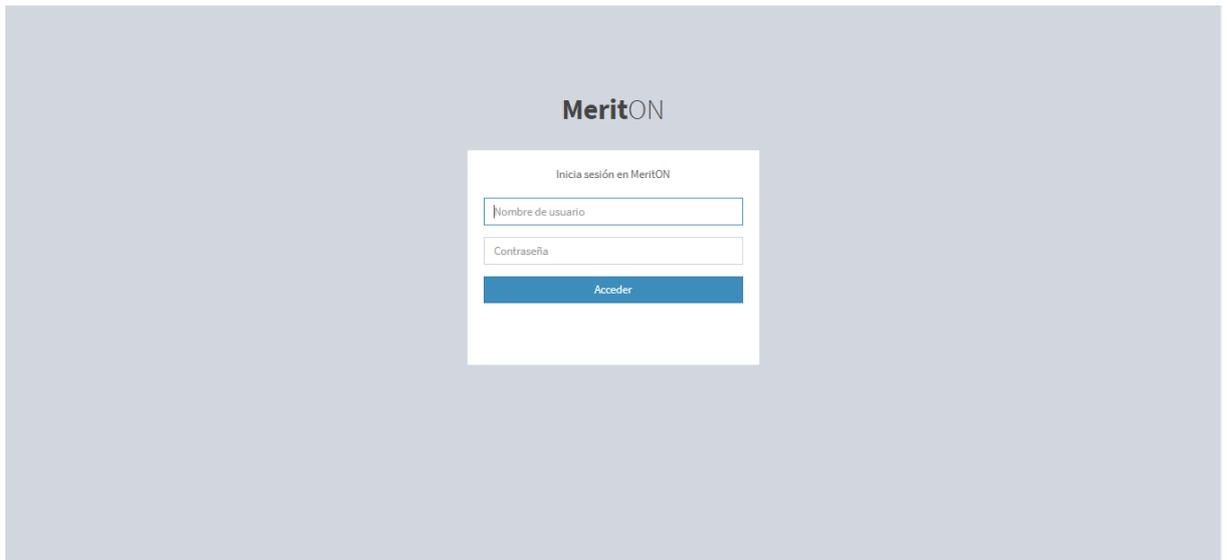


Ilustración 33. Vista de Login de usuario

Para poder acceder a la aplicación es necesario que las credenciales introducidas sean validadas, por lo que para ello se creará un controlador que se encargue de validar los campos del formulario login.

Este controlador, comprueba que el alias y la contraseña, previamente encriptada mediante una librería de Codeigniter, se encuentren en la base de datos y coincidan.

5.1.4. Pruebas

En esta fase de pruebas, el usuario deberá de ser capaz de iniciar sesión utilizando un alias y una contraseña que previamente ha introducido el administrador en la base de datos, ya que aún no está creada el formulario de crear usuarios.

El resultado se muestra en las ilustraciones siguientes:

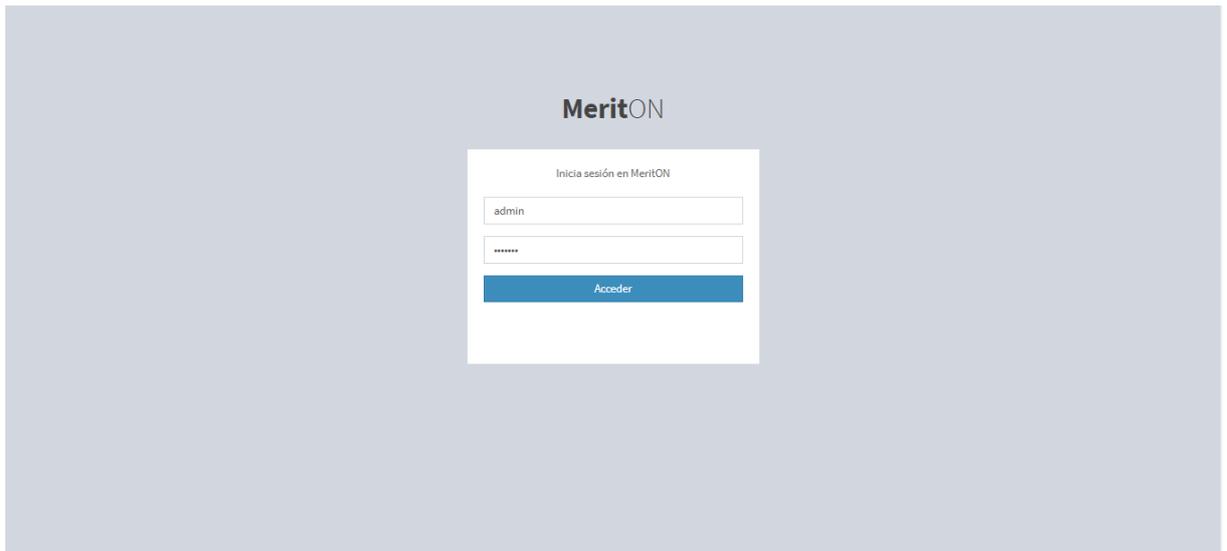


Ilustración 34. Introducir credenciales login

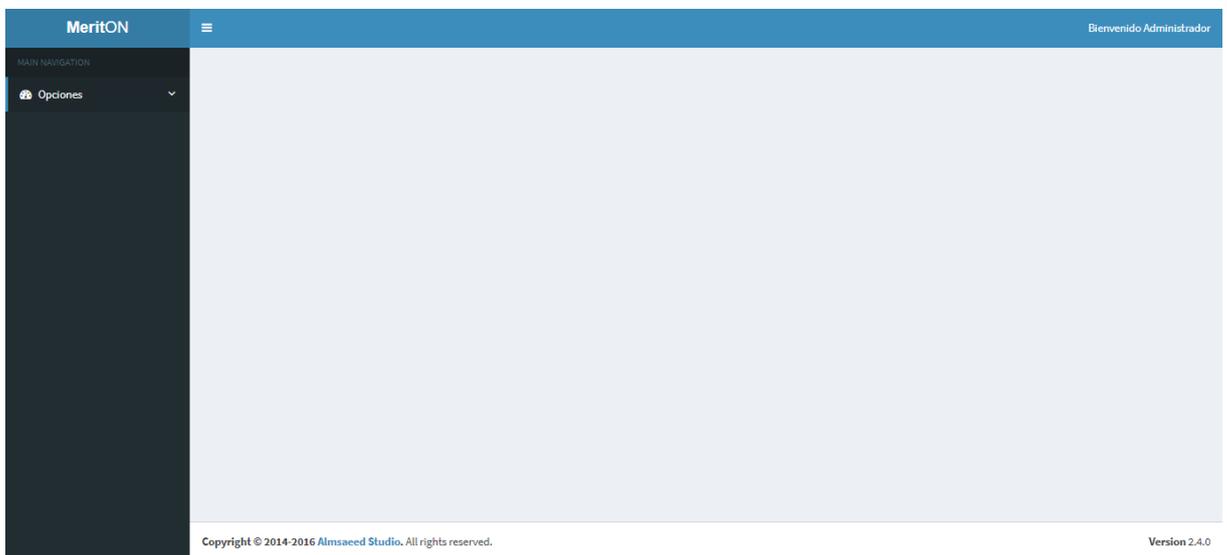


Ilustración 35. Vista de Inicio usuario logueado

5.2. Segundo Incremento

En este segundo incremento se va a proceder a realizar las vistas y controladores de la gestión de usuarios, las cuales van a ser accesibles solo por el administrador.

5.2.1. Análisis

5.2.1.1. Caso de Uso

En la Ilustración 36, se puede apreciar los casos de uso sobre la gestión de usuarios. En este diagrama, se observa que el administrador, es el único capaz de

crear a los usuarios, así como modificarlos y darlos de baja. Bajo el rol de profesor, se pueden relacionar a los tutores legales con los alumnos, pero eso se realizará en próximos incrementos.

Todos los usuarios tienen la opción de iniciar sesión y cerrarla, los privilegios de modificar los datos, son solo accesibles por el administrador.

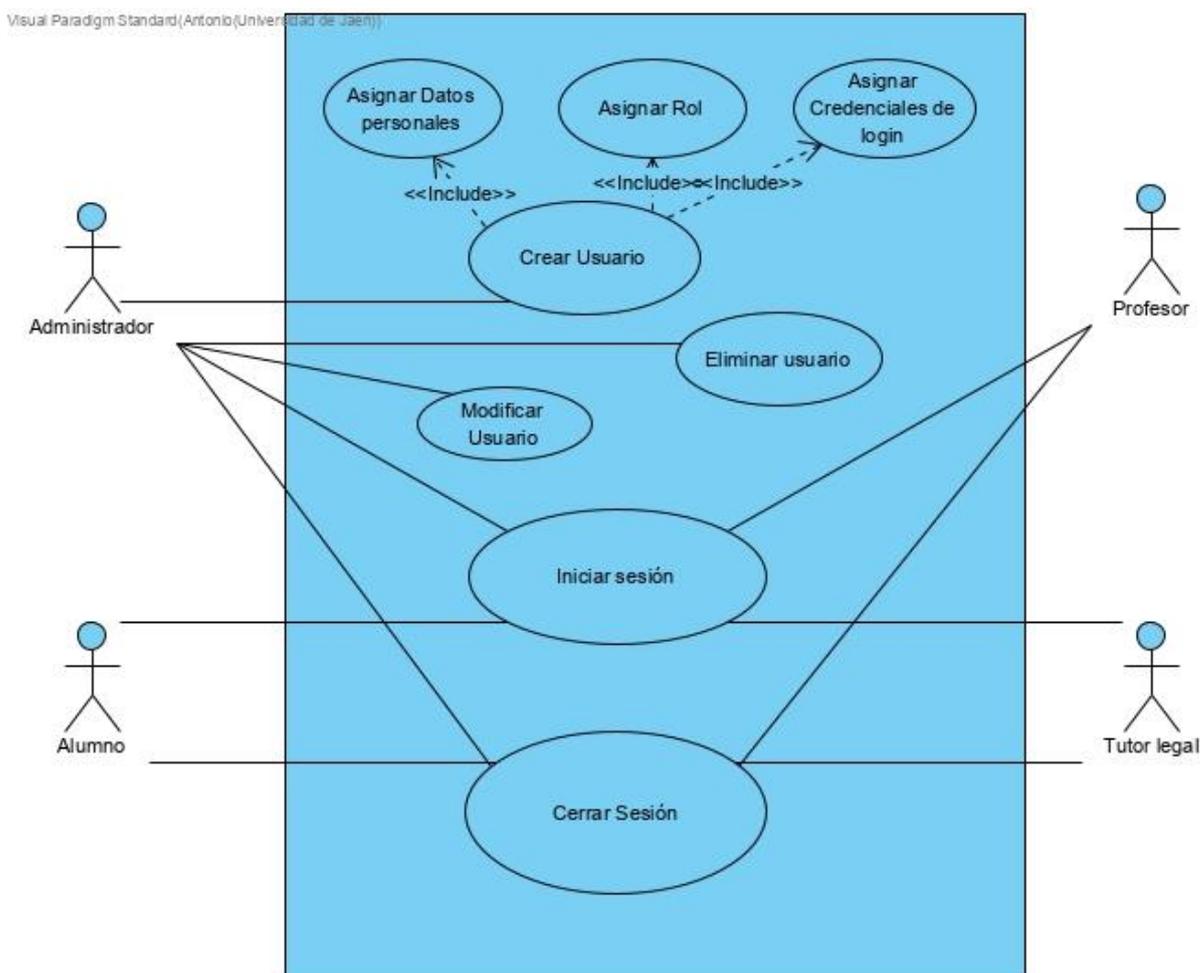


Ilustración 36. Diagrama de casos de uso de la gestión de los usuarios.

5.2.2. Diseño

5.2.2.1. Diagrama de Secuencia

En el siguiente diagrama, se muestra la creación de un usuario por parte del administrador.

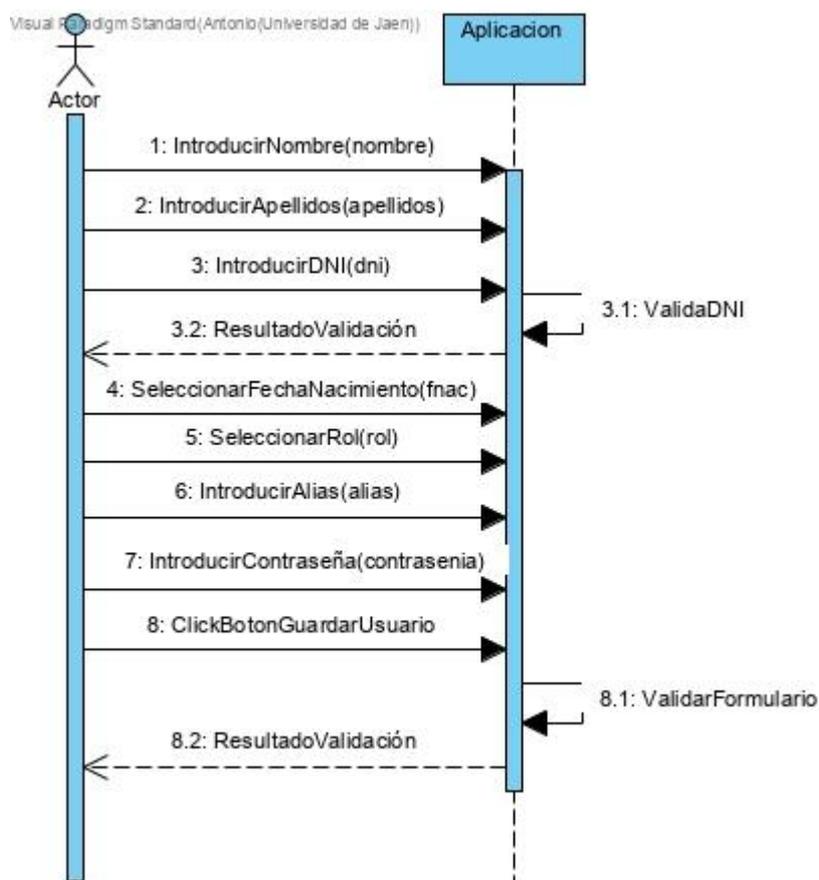


Ilustración 37. Diagrama de secuencia creación de usuario.

Para acceder a este formulario, habría que iniciar sesión como administrador, el diagrama de secuencia asociado al inicio de sesión ya se ha explicado anteriormente en la Ilustración 20.

5.2.2.2. Interfaz de Usuario

Con respecto a este apartado, se seguirá el storyboard sobre la creación de perfil mostrado en el incremento anterior. Ilustración 22.

5.2.3. Implementación

Se procederá a crear la vista en la que aparece tanto el formulario de creación de usuarios, el primer paso es crear el archivo en la carpeta views del proyecto. El siguiente paso, es realizar en el modeloUsuario creado en el anterior incremento, los métodos necesarios para introducir los datos en la base de datos, y poder actualizarlos. Estos métodos serán llamados a través de un controlador que vamos a crear para comunicarnos con la vista, el controlador se llama controladorUsuario, y,

obteniendo los datos introducidos en la vista, se los envía al modelo, mediante un vector en el que han sido guardados dichos datos.

En esta vista, se ha incluido un formulario, en el que se permite realizar una búsqueda del usuario que requiera el administrador, mediante la introducción de su DNI, a partir de ahí puede cambiar sus datos.

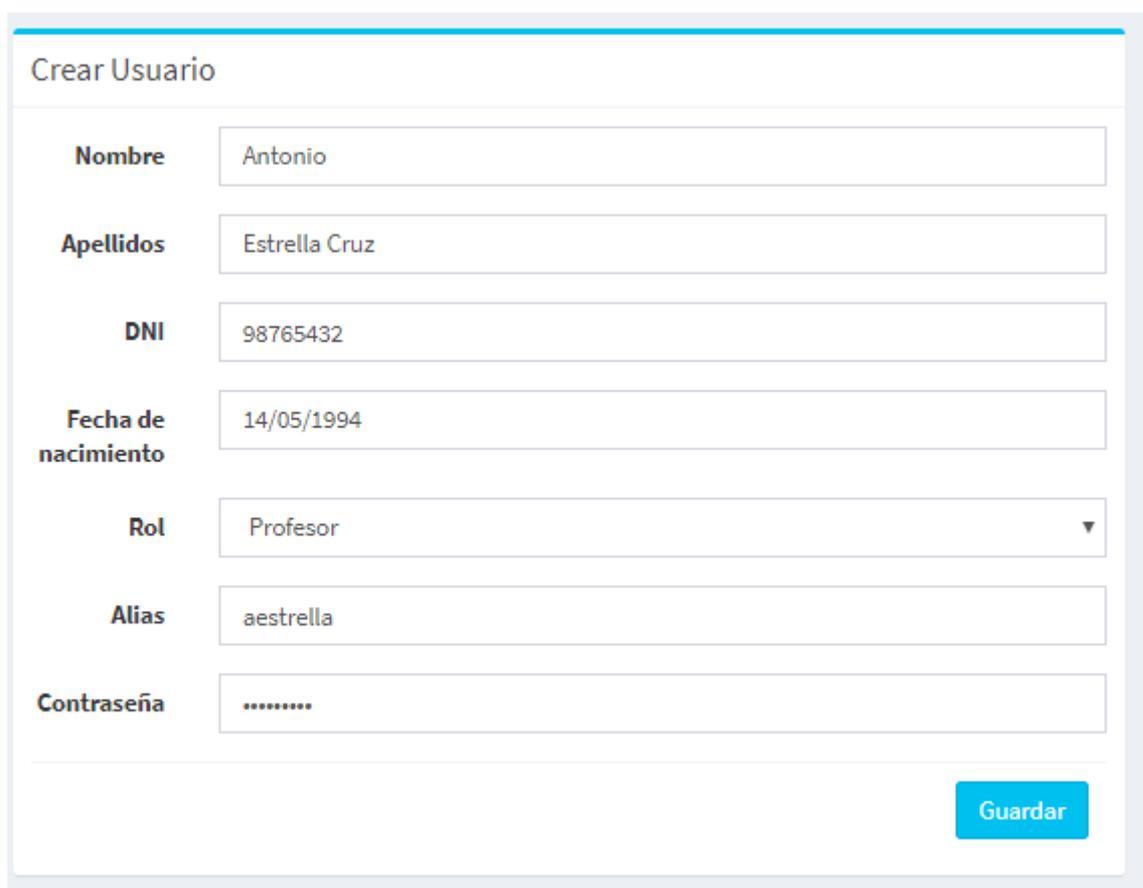
Copyright © 2014-2016 Almsaeed Studio. All rights reserved. Version 2.4.0

Ilustración 38. Formulario de creación de usuarios y actualización de usuarios (administrador).

Como se puede observar en la Ilustración 38, dentro del formulario existe un desplegable, este desplegable corresponde a la elección del rol que el usuario va a ejercer dentro de nuestra aplicación, el cual puede ser, Profesor, Alumno, O tutor legal, como se ha mencionado anteriormente.

5.2.4. Pruebas

Para este incremento, la fase de pruebas va a consistir en crear un usuario de forma correcta y comprobar que el usuario ha sido introducido en nuestra base de datos.



The screenshot shows a web form titled "Crear Usuario" with the following fields and values:

Field	Value
Nombre	Antonio
Apellidos	Estrella Cruz
DNI	98765432
Fecha de nacimiento	14/05/1994
Rol	Profesor
Alias	aestrella
Contraseña

A blue "Guardar" button is located at the bottom right of the form.

Ilustración 39. Pruebas crear usuario

En la Ilustración 39, se puede observar los datos que se van a introducir, y en la Ilustración 40, los datos correctos introducidos en la base de datos, el primer dato correspondería al id único del usuario, generado automáticamente, el segundo dato corresponde a la contraseña, que ha sido encriptada mediante una librería de codeigniter, los siguientes datos se tratan del nombre, apellidos, fecha de nacimiento, tipo de usuario (1 en este caso corresponde a profesor), DNI sin letra, y el alias con el que accedería el usuario a la aplicación.

✖ Editar 📄 Copiar 🗑 Borrar 15 f7c3bc1d808e04732adf679965ccc34ca7ae3441 Antonio Estrella Cruz 1994-05-14 1 98765432 aestrella

Ilustración 40. Usuario introducido correctamente en la base de datos

5.3. Tercer incremento

En este incremento se va a proceder a realizar la funcionalidad relacionada con la creación de insignias y las relaciones entre usuarios, tutores legales con alumnos y profesores con sus respectivos alumnos.

5.3.1. Análisis

5.3.1.1. Casos de Uso

Como se ha realizado en los anteriores incrementos, el primer paso es realizar el diagrama de casos de uso, en este incremento, se van a realizar dos, uno para la creación de insignias, y otro para la creación de las relaciones entre usuarios.

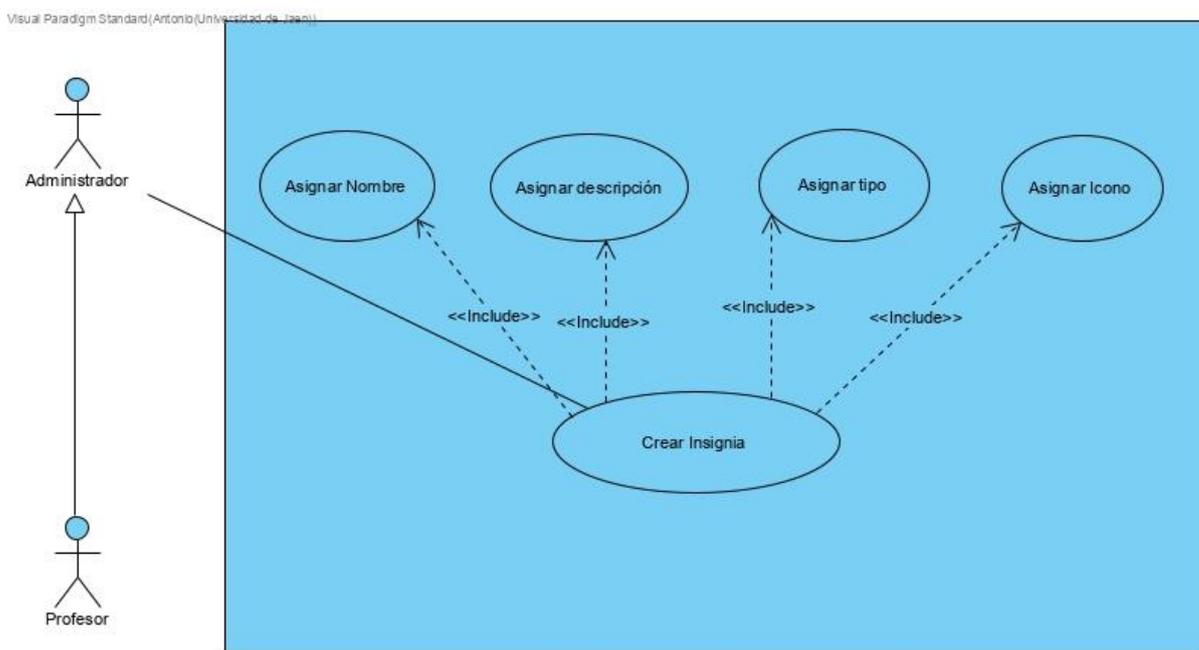


Ilustración 41. Diagrama de caso de uso de creación de insignias

Analizando el diagrama de la Ilustración 41, se puede concluir que tanto el administrador como el usuario tipo profesor son los usuarios encargados de crear las insignias que posteriormente se van a asignar a los alumnos.

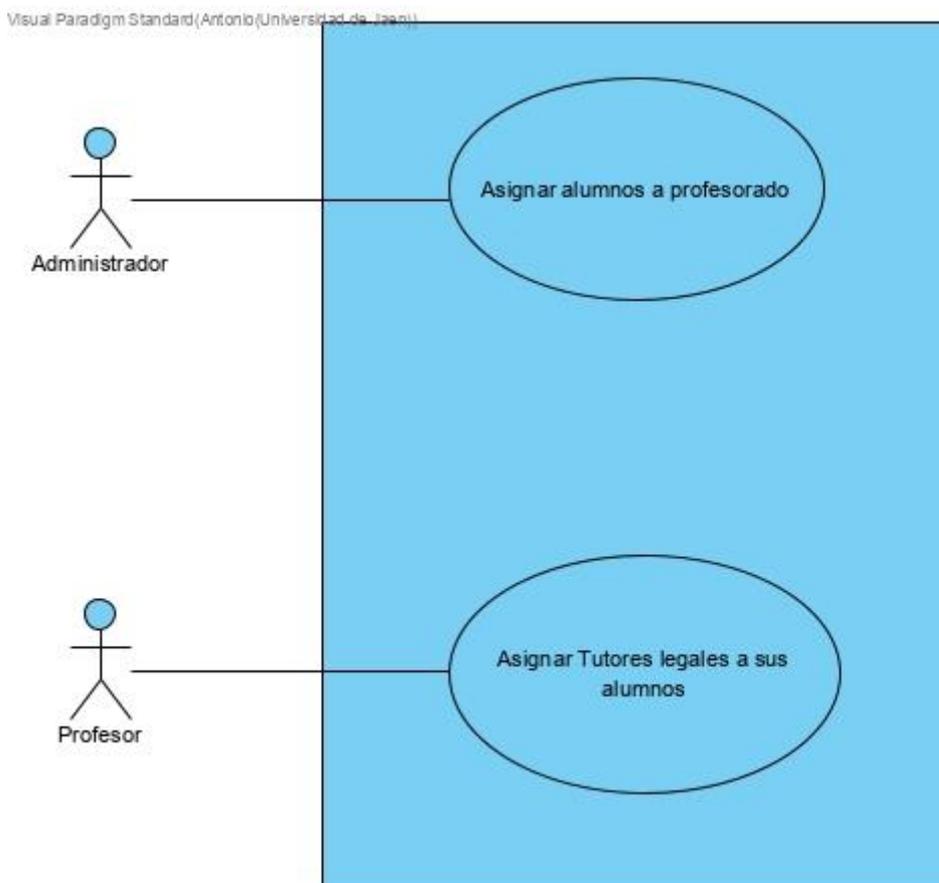


Ilustración 42. Diagrama de caso de uso sobre la relación entre usuarios.

En este diagrama se muestra que el encargado de asignar a cada profesor sus alumnos correspondientes es el administrador, y el encargado de asignar a cada alumno sus tutores legales, es el profesor.

5.3.2. Diseño

Para la interfaz se va a proceder a realizar tres vistas. La primera vista va a ser la vista en la cual se muestra el formulario de creación de insignias, y aparte contará con un visor de dichas insignias. Ilustración 43

La segunda vista corresponde a la asignación de los alumnos a sus respectivos profesores, por lo que se mostrará una tabla en la que aparece un listado con todos los alumnos que figuran en la base de datos, una de las celdas correspondientes a cada alumno, se trata de un botón para asignar a los alumnos. Al pulsar dicho botón, aparece una ventana emergente, en la que se encuentra el nombre del alumno y un desplegable en el cual aparecen todos los profesores incluidos en la base de datos. Ilustración 44.

La tercera vista es similar a la segunda, solo que esta vista es solo accesible por los profesores. A cada profesor le aparece un listado de los alumnos que tiene asociados, como se ha expuesto anteriormente, en la ventana emergente aparece un desplegable con los distintos tutores legales existentes. Ilustración 44.

5.3.2.1. Diagramas de secuencia

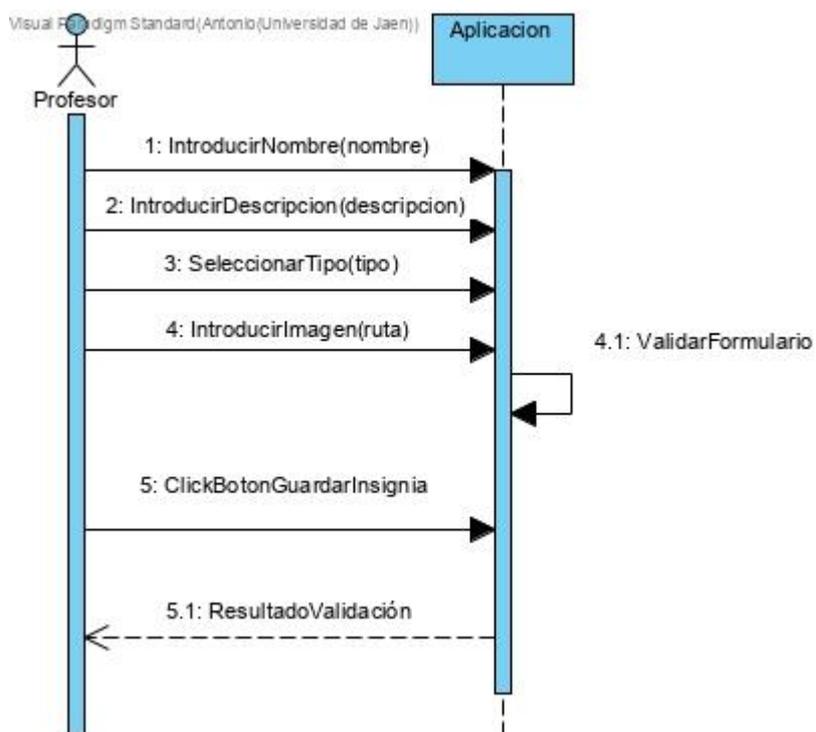


Ilustración 43. Diagrama de Secuencia de crear insignias.

En la Ilustración 43, se observa el proceso de creación de una insignia, el usuario profesor debe introducir los datos de la insignia nombre, descripción, tipo de insignia y la correspondiente imagen, el servidor valida el formulario, para comprobar que todos los datos son correctos.

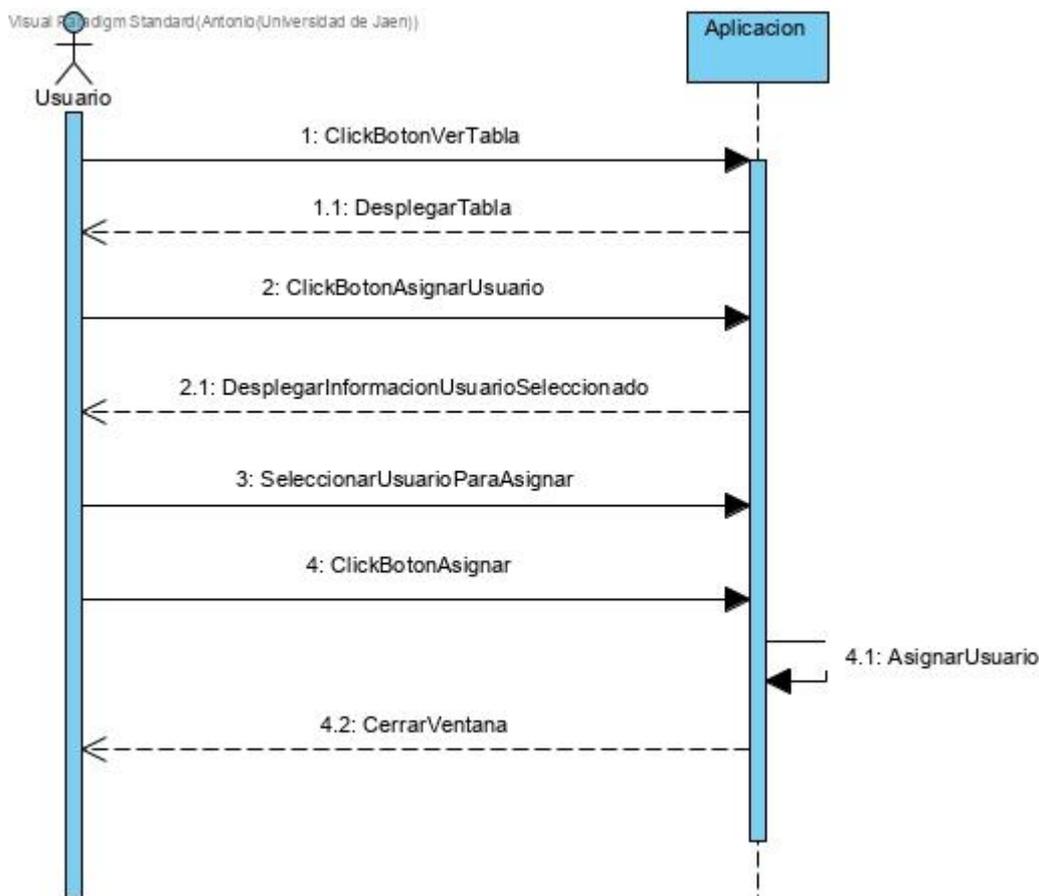


Ilustración 44. Diagrama de Secuencia para relacionar usuarios.

La relación de los usuarios se puede llevar a cabo desde el tipo de usuario profesor y el tipo de usuario administrador, para relacionar un alumno con un profesor, el administrador hace click en la opción de ver la tabla de alumnos totales, selecciona la opción de asignar un profesor a un alumno. Al seleccionar esta opción aparece una ventana emergente en la cual se muestra un desplegable con la lista de profesores.

Para relacionar a los alumnos con los tutores legales, sería el mismo proceso, pero con tutores legales en vez de profesores, y la lista de alumnos que aparecería es solo la que tiene asignada el profesor encargado de la asignación.

5.3.2.2. Interfaz de Usuario

Con respecto a este apartado, se seguirá el storyboard sobre la creación de insignias y relación de usuarios. Ilustración 24. Panel de creación de Insignias, Ilustración 23 e Ilustración 26. Vista de asignación de tutores legales.

5.3.3. Implementación

Se procederá a crear primero la vista de creación de insignias. Dicha vista será accesible solamente por los profesores y administradores y estará compuesta por un formulario en el cual se introducen los datos de cada insignia, junto con una imagen de identificación de la misma. También cuenta con un apartado dentro de la vista en la que se muestran las insignias que se encuentran en la base de datos.

El primer paso para crear la vista, es crear un controlador que relacione la vista con `modeloInsignias`. En este modelo se van a implementar los métodos necesarios para introducir los datos de las insignias en la base de datos.

El controlador será el encargado de enviar los datos introducidos en la vista a `modeloInsignias` mediante un array. Como se ha expuesto anteriormente, uno de los campos del array es un archivo de imagen, en el controlador se ha implementado un método, el cual, al guardar la imagen, la redimensiona y crea una miniatura de la misma, gracias a una librería que existe en codeigniter. Esta librería permite cambiar la configuración de la subida de imágenes, con los parámetros que se muestran en la siguiente Ilustración 45.

Preference	Default Value	Options	Description
<code>upload_path</code>	None	None	The path to the directory where the upload should be placed. The directory must be writable and the path can be absolute or relative.
<code>allowed_types</code>	None	None	The mime types corresponding to the types of files you allow to be uploaded. Usually the file extension can be used as the mime type. Can be either an array or a pipe-separated string.
<code>file_name</code>	None	Desired file name	If set CodeIgniter will rename the uploaded file to this name. The extension provided in the file name must also be an allowed file type. If no extension is provided in the original <code>file_name</code> will be used.
<code>file_ext_tolower</code>	FALSE	TRUE/FALSE (boolean)	If set to TRUE, the file extension will be forced to lower case
<code>overwrite</code>	FALSE	TRUE/FALSE (boolean)	If set to true, if a file with the same name as the one you are uploading exists, it will be overwritten. If set to false, a number will be appended to the filename if another with the same name exists.
<code>max_size</code>	0	None	The maximum size (in kilobytes) that the file can be. Set to zero for no limit. Note: Most PHP installations have their own limit, as specified in the <code>php.ini</code> file. Usually 2 MB (or 2048 KB) by default.
<code>max_width</code>	0	None	The maximum width (in pixels) that the image can be. Set to zero for no limit.
<code>max_height</code>	0	None	The maximum height (in pixels) that the image can be. Set to zero for no limit.
<code>min_width</code>	0	None	The minimum width (in pixels) that the image can be. Set to zero for no limit.
<code>min_height</code>	0	None	The minimum height (in pixels) that the image can be. Set to zero for no limit.
<code>max_filename</code>	0	None	The maximum length that a file name can be. Set to zero for no limit.

Ilustración 45. Opciones de la librería para subir imágenes

Ilustración 46. Formulario de creación de Insignias.

A continuación, se procederá a crear las dos vistas de relación de usuarios. Ambas vistas comparten diseño, pero el contenido desplegado en ambos es distinto. Primero, se va a hacer uso del controlador de usuarios creado previamente. En este controlador se van a crear dos métodos get, que obtienen de la BD, los datos de los usuarios que se van a mostrar en la tabla, y los datos de los usuarios que aparecerán en el desplegable para relacionarlos.

ID Alumno	Nombre	Apellidos	Alias	Fecha de nacimiento	Asignar Profesores
16	Antonio	Herrera Melero	adiaz	1995-07-20	 Asignar profesor

Ilustración 47. Tabla de asignación de profesores



Ilustración 48. Vista de la ventana emergente para asignar profesores

5.3.4. Pruebas

Para este incremento la fase de pruebas va a consistir en crear una insignia, comprobar que se ha introducido de forma correcta, y relacionar a los usuarios, tanto a los profesores con sus alumnos como con los tutores legales.

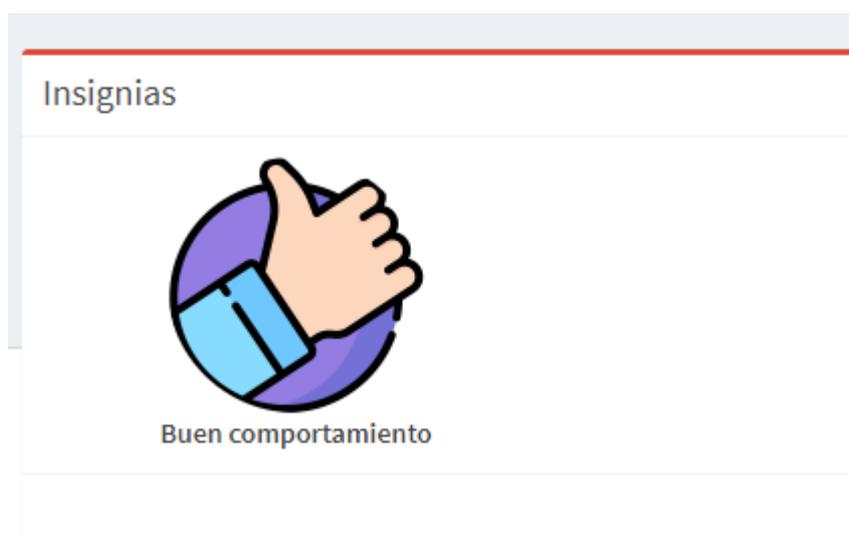


Ilustración 49. Insignia creada

Listado de Alumnos de Antonio Estrella Cruz

Show entries Search:

ID Alumno	Nombre	Apellidos	Alias	Fecha de nacimiento	Asignar Insignias
16	Antonio	Herrera Melero	adiaz	1995-07-20	<input checked="" type="checkbox"/> Asignar Insignias

Showing 1 to 1 of 1 entries Previous **1** Next

Ilustración 50. Listado de alumnos asignados a un profesor

5.4. Cuarto Incremento

En este cuarto y último incremento se van a realizar las vistas y funcionalidades que faltan por implementar, correspondientes al resto de usuarios (tutores legales y alumnos), y la vista de asignación de insignias a cada alumno.

5.4.1. Análisis

5.4.1.1. Diagrama de Casos de Uso

Como se ha hecho en el resto de incrementos, creare el caso de uso correspondiente a las funcionalidades que se implementaran en este incremento.

En la Ilustración 51, se muestra como el profesor es el encargado de asignar las insignias a sus propios alumnos.

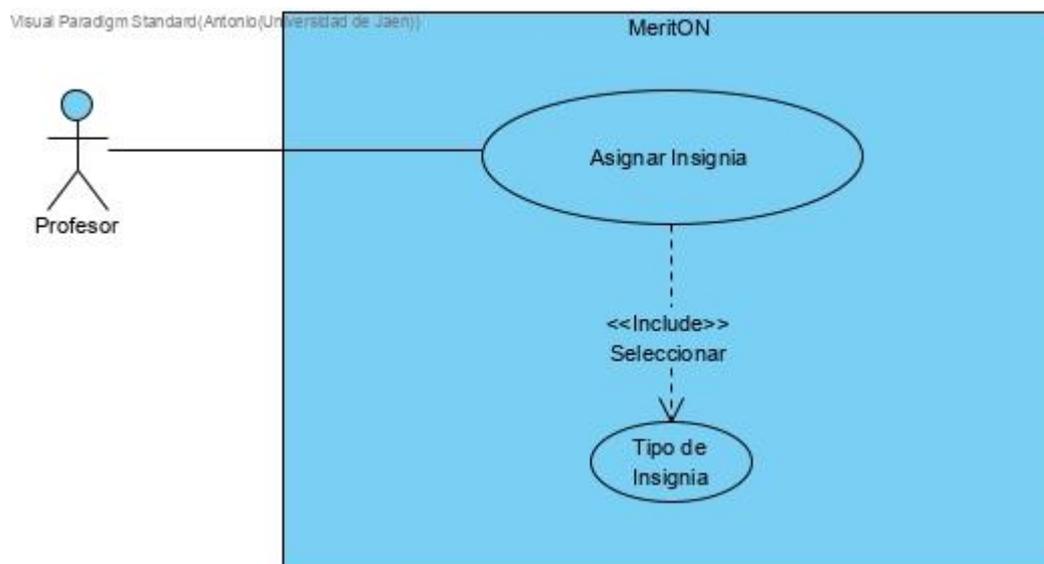


Ilustración 51. Diagrama de Caso de Uso de asignación de insignias

También se llevará a cabo la implementación de los casos de uso de la Ilustración 19, que corresponde con la visualización de las vistas principales de cada usuario, teniendo en cuenta que según el rol del usuario logueado, la vista principal mostrará diferente información.

5.4.2. Diseño

5.4.2.1. Diagrama de secuencia

El diagrama de secuencia mostrado a continuación corresponde con la interacción que hace el usuario profesor con el sistema al asignar insignia.

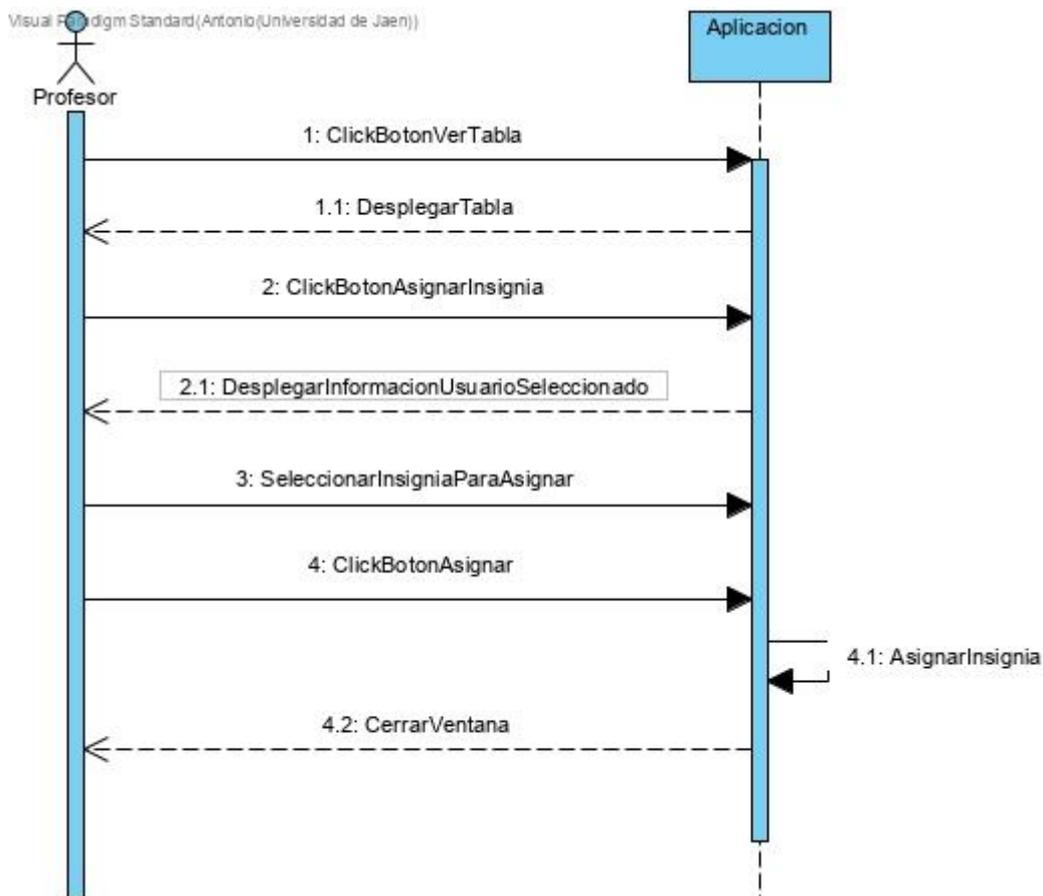


Ilustración 52. Diagrama de secuencia asignación de insignias.

Para asignar una insignia el profesor se ha utilizado una implementación muy parecida a la de relación de usuarios. El profesor accede a su listado de alumnos, en la tabla de alumnos, aparece una opción en cada alumno que al hacer click, muestra una ventana emergente, en dicha ventana aparecen los datos del alumno y un

desplegable con las insignias que hay creadas, por lo que el profesor selecciona la insignia y pulsa el botón de asignar.

5.4.2.2. Interfaz de usuario

Con respecto a la interfaz de usuario de la aplicación, las vistas que faltarían por implementar, son la de asignación de insignias, y las vistas personales de los usuarios tipo alumno y tipo tutor legal.

5.4.3. Implementación

Para implementar estas vistas, se han introducido nuevos métodos en los controladores que ya habían sido creados para las otras vistas.

En primer lugar, para la vista mediante la cual el profesor puede asignar las insignias, se ha realizado una consulta a la base de datos por la cual obtendríamos una tabla con los alumnos asignados a ese profesor, en la tabla aparece al lado de cada alumno una nueva celda, con una opción de asignar insignias. Al hacer click en esta opción, aparece una ventana emergente con los datos del alumno, y un panel seleccionable con las insignias disponibles en la base de datos.



Listado de Alumnos de Antonio Estrella Cruz

Show 10 entries Search:

ID Alumno	Nombre	Apellidos	Alias	Fecha de nacimiento	Asignar Insignias
16	Antonio	Herrera Melero	adiaz	1995-07-20	 Asignar Insignias

Showing 1 to 1 of 1 entries Previous 1 Next

Mostrar

Ilustración 53. Listado de alumnos a los que asignarles insignias.

Asignación de Insignias



Antonio Herrera Melero

Seleccionar Insignia

Ilustración 54. Vista de asignación de insignias.

Para la vista principal de los alumnos, se ha creado un método en `modeloInsignias`, que devuelve de la base de datos las insignias asignadas a ese alumno solamente de tipo positivo. Ilustración 55

Por otra parte, con respecto a la vista principal de los tutores legales, se ha creado un método similar en `modeloInsignias`, pero con la variante de que devuelve todos los tipos de insignias, tanto las positivas como las negativas. Aparte se ha creado otro método en el `modeloUsuario`, que obtiene todos los alumnos que tienen relación con ese tutor legal.

La vista final consiste en una tabla con los alumnos que tiene relacionados cada tutor legal, y una opción que, al pulsar sobre ella, muestra las insignias tanto positivas como negativas de cada alumno.

5.4.4. Pruebas

Para la fase de pruebas, se han asignado varias insignias de distinto tipo a una serie de alumnos cuyo tutor legal es el mismo, para comprobar que la asignación ha sido correcta. Poder ver en el usuario alumno que las insignias asignadas son positivas. Por último, en la vista de tutor legal, realizar una comprobación sobre las insignias que corresponden a cada alumno que tenga asignado, tanto positivas como negativas.

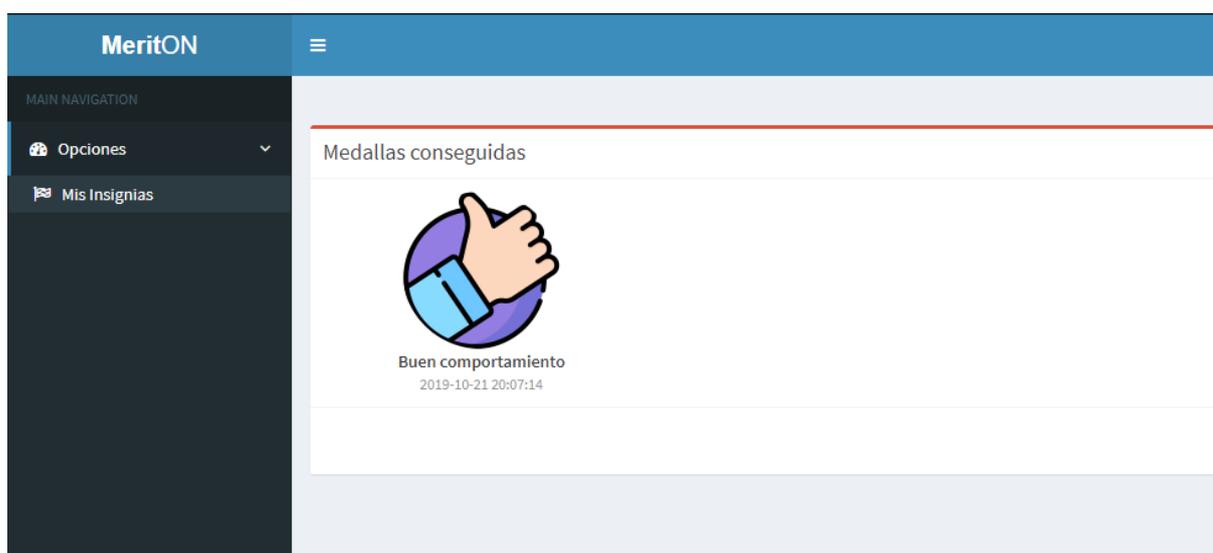


Ilustración 55. Vista principal del alumno.

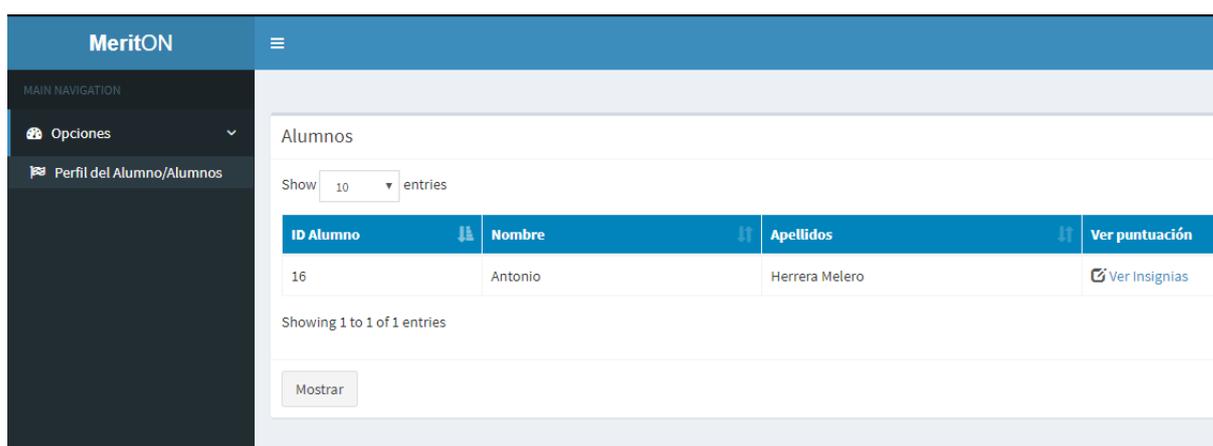


Ilustración 56. Vista principal del tutor legal.

5.5. Revisión del prototipado

El prototipo ha sido revisado por los docentes previamente consultados en primera instancia, por lo que han podido comprobar el funcionamiento, y recomendar algunas mejoras, las cuales se muestran en el siguiente capítulo.

6. CONCLUSIONES Y MEJORAS FUTURAS

Una vez finalizado el prototipo de la aplicación, se han obtenido las siguientes conclusiones:

- La aplicación funciona también para dispositivos móviles debido a que se ha creado de manera responsive, gracias a la plantilla elegida.
- El mayor problema encontrado ha sido a la hora de relacionar las tablas en la base de datos. Todos los tipos de usuarios comparten una misma tabla, por lo que se han tenido que crear nuevas tablas para crear las distintas relaciones.
- La arquitectura usada ha sido una gran ventaja en este proyecto, ya que como se ha comentado anteriormente, el patrón MVC es muy sencillo de implementar, y existe una gran variedad de información.
- Con respecto a las tecnologías utilizadas, todas ellas disponen de una gran cantidad de documentación, en especial CodeIgniter y PHP, que disponen de páginas oficiales.
- Como conclusión final, podemos observar la gran aplicación que tendría este proyecto, tanto en instituciones públicas como privadas, y en distintos niveles de la educación.

A continuación, paso a detallar las mejoras futuras de la aplicación:

- Un sistema de juegos por el cual el alumno puede obtener automáticamente al finalizar el juego una nueva insignia. En estos juegos todos los alumnos obtendrían alguna insignia, simplemente por haber participado en el juego, para que todos tengan la misma oportunidad de obtener insignias.
- Incorporación de un sistema de gráficos para los profesores, para que puedan ver la evolución que van teniendo sus alumnos, y así poder reforzar los puntos débiles.
- Incorporar un sistema de chat con los tutores legales de los alumnos, que comunique al profesor directamente con ellos, y les pueda comunicar los aspectos a reforzar en casa de los alumnos.

- Un sistema de evaluación, por el cual el profesor crea una serie de preguntas que aparecen en pantalla y los alumnos desde su dispositivo e iniciados con su cuenta, puedan seleccionar la respuesta correcta.
- Creación de un nuevo campo en la tabla de usuarios. Dicho campo es una imagen de identificación del usuario, en el caso de los alumnos, se debería crear aparte un editor de imágenes mediante el cual los alumnos puedan personalizar su icono representativo. Los usuarios de tipo tutor legal y profesor si contarían con una foto personal.

ANEXO I. MANUAL DE INSTALACIÓN

Se va a detallar el proceso de instalación que se ha de llevar a cabo para la utilización de la aplicación. El primer paso es instalar en nuestro sistema el software descrito anteriormente XAMPP, el cual es gratuito, para ello se descarga de la página oficial y se elige el SO en el que se va realizar la instalación.



Ilustración 57. Elección de Xampp según SO.

Como siguiente paso, se procederá a realizar la instalación, en el que se elegirá donde se va a instalar. Como recomendación, lo ideal es avanzar en todos los pasos de la instalación con las opciones asignadas por defecto, ya que es suficiente para los requisitos de la aplicación.

Tras realizar la instalación, se van a introducir los ficheros para la base de datos y para la aplicación web.

Base de datos

Una vez iniciado el programa Xampp, se debe activar las opciones de Apache y MySQL en el panel de control que nos muestra el software. Ilustración 58

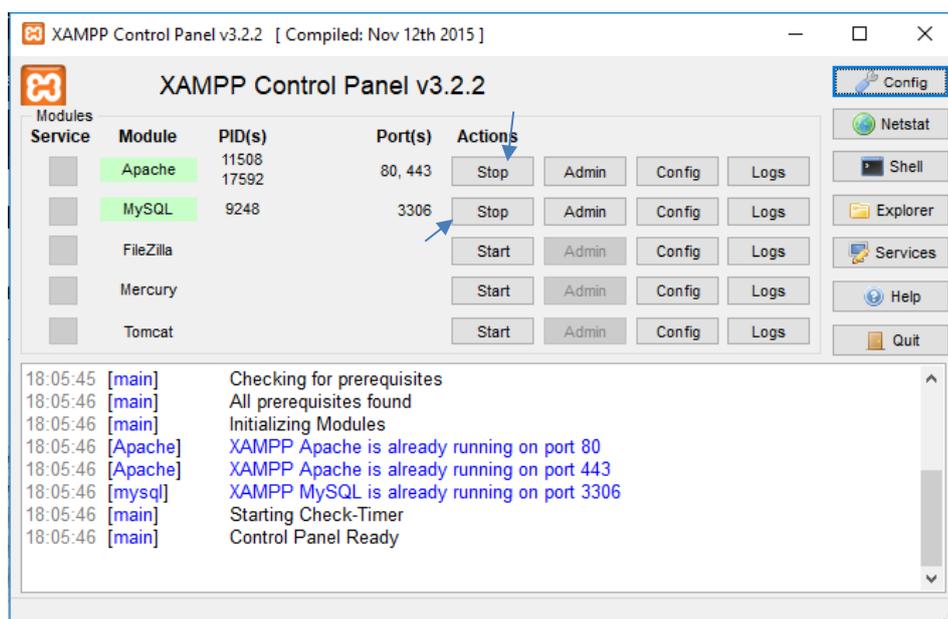


Ilustración 58. Apache y MySQL activados.

A continuación, es necesario acceder al panel de control de la base de datos introduciendo en el navegador la url: localhost/phpmyadmin. Dentro del panel de control, se va a proceder a crear una base de datos nueva llamada tfg.

Una vez creada, se procederá a importar el archivo tfg.sql haciendo click sobre la opción importar, y seleccionando el archivo citado anteriormente. Las opciones que aparecerán se dejarán por defecto.

Aplicación Web

Para la instalación de la aplicación, el primer cometido es incluir el proyecto dentro de la carpeta htdocs que se ha generado al instalar Xampp.

La localización de la carpeta si se ha instalado por defecto es C:/xampp/htdocs. En esta carpeta es donde vamos a copiar la carpeta TFG.

Por último, para hacer uso de la aplicación es necesario iniciar el programa Xampp, con las opciones Apache y MySQL activadas, como se ha descrito anteriormente. La url que hay que introducir en el navegador para utilizar la aplicación es: <http://localhost/TFG/>.

ANEXO II. MANUAL DE USUARIO

El software está implementado de forma local, por lo tanto, el primer paso después de haber instalado XAMPP, es iniciar el programa y activar los módulos de Apache y MySQL.

Para acceder a la aplicación web, hay que introducir la siguiente dirección en el navegador web: localhost/TFG y nos direccionaría a la página de inicio como se puede apreciar en la Ilustración 59

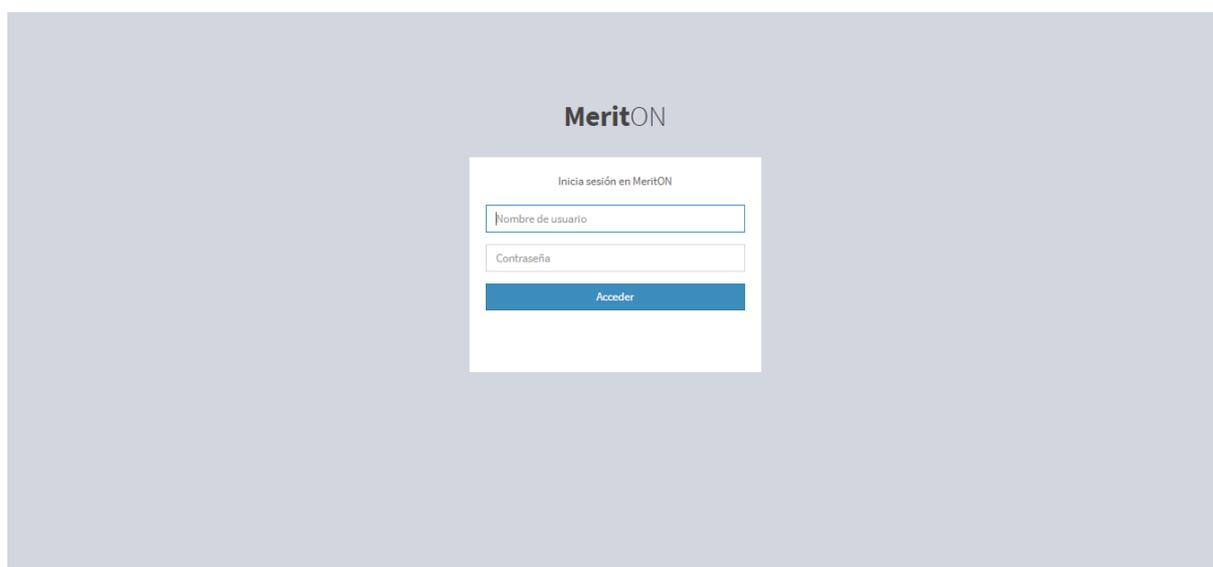


Ilustración 59. Inicio de sesión

Para acceder a la aplicación, debido a que existen cuatro tipos de usuarios distintos, se ha creado un usuario por cada rol. El alias de inicio y las contraseñas son:

- Administrador: alias (admin), contraseña (admin)
- Profesor: alias (aestrella), contraseña (123456789)
- Alumno: alias (adiaz), contraseña (123456)
- Tutor legal: alias(jcabeza), contraseña(1234)

Las opciones que tienen los usuarios son diferentes según su perfil. En la tabla siguiente se muestran cada una de las opciones y el usuario que puede acceder a ellas.

	Administrador	Profesor	Alumno	Tutor legal
Consultar todas las insignias asignadas	Sí	Sí	No	Sí
Consultar insignias de tipo positivo	Sí	Sí	Sí	Sí
Crear usuarios	Sí	No	No	No
Crear Insignias	Sí	Sí	No	No
Asignar insignias	Sí	Sí	No	No
Relacionar Profesores con Alumnos	Sí	No	No	No
Relacionar Tutores legales con Alumnos	Sí	Sí	No	No

Tabla 7. Opciones del rol

Consultar todas las insignias asignadas

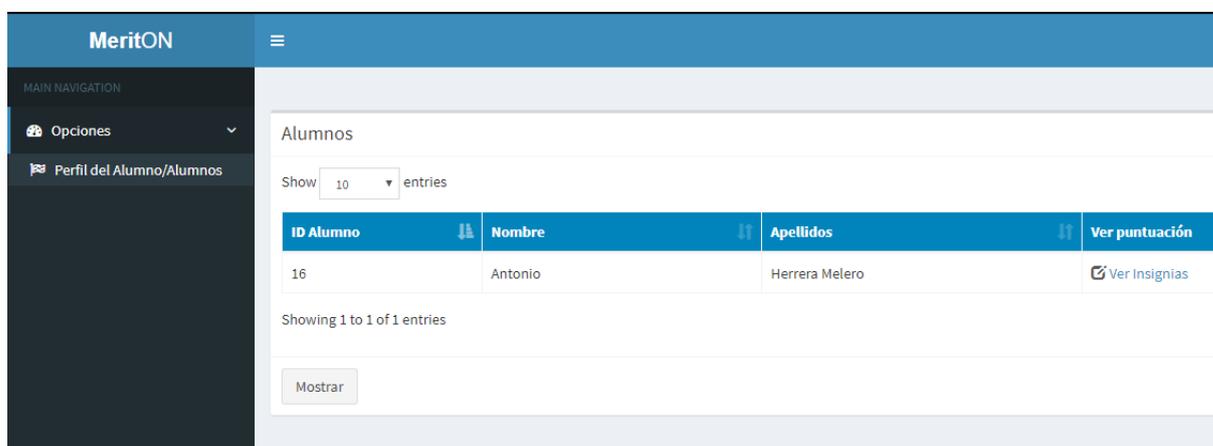
Para poder consultar las insignias asignadas, el administrador puede acceder a los perfiles de todos los alumnos, mediante un desplegable en el que se muestra una tabla con todas las insignias asignadas a los alumnos.

El profesor puede realizar la misma acción pero solamente puede ver las insignias que ha asignado a sus alumnos.

El tutor legal puede observar todas las insignias que han sido asignadas a los alumnos los cuales tutoriza.

El alumno solamente puede ver las insignias positivas que le han sido asignadas.

Todos ellos lo hacen mediante la opción ver Insignias asignadas, al hacer click en ella se despliega una tabla en la que aparecen los datos de los alumnos, y una opción que pulsando sobre ella, mostraría mediante una ventana emergente las insignias que le han sido asignadas. Ilustración 60



The screenshot shows the MeritON interface. On the left is a dark navigation sidebar with 'Opciones' and 'Perfil del Alumno/Alumnos'. The main content area is titled 'Alumnos' and features a table with the following data:

ID Alumno	Nombre	Apellidos	Ver puntuación
16	Antonio	Herrera Melero	Ver Insignias

Below the table, it says 'Showing 1 to 1 of 1 entries' and there is a 'Mostrar' button.

Ilustración 60. Vista principal de la tabla con la opción de ver insignias.

Consultar insignias de tipo positivo

Esta opción es la que se da cuando un alumno accede a la aplicación a través de su cuenta, aunque el tenga asignadas insignias negativas solamente podrá ver las positivas.

Esta opción es muy sencilla de observar, simplemente el alumno deberá acceder con su usuario y pulsando sobre la opción “Ver mis medallas” podrá tener acceso a sus insignias.

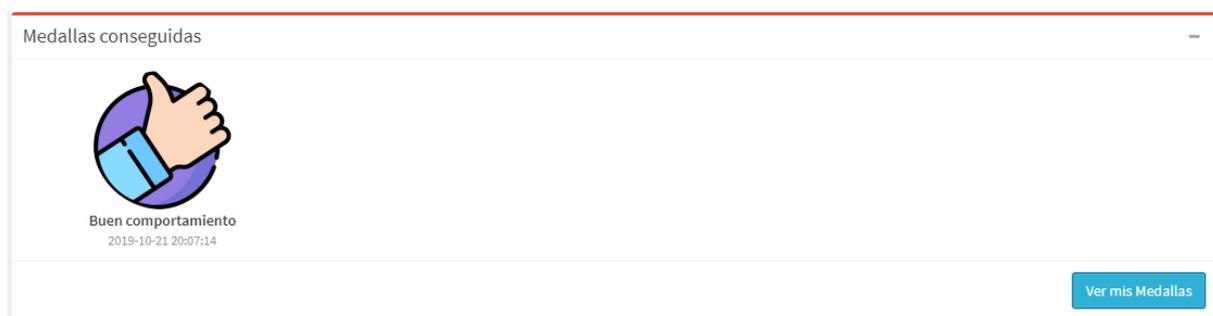


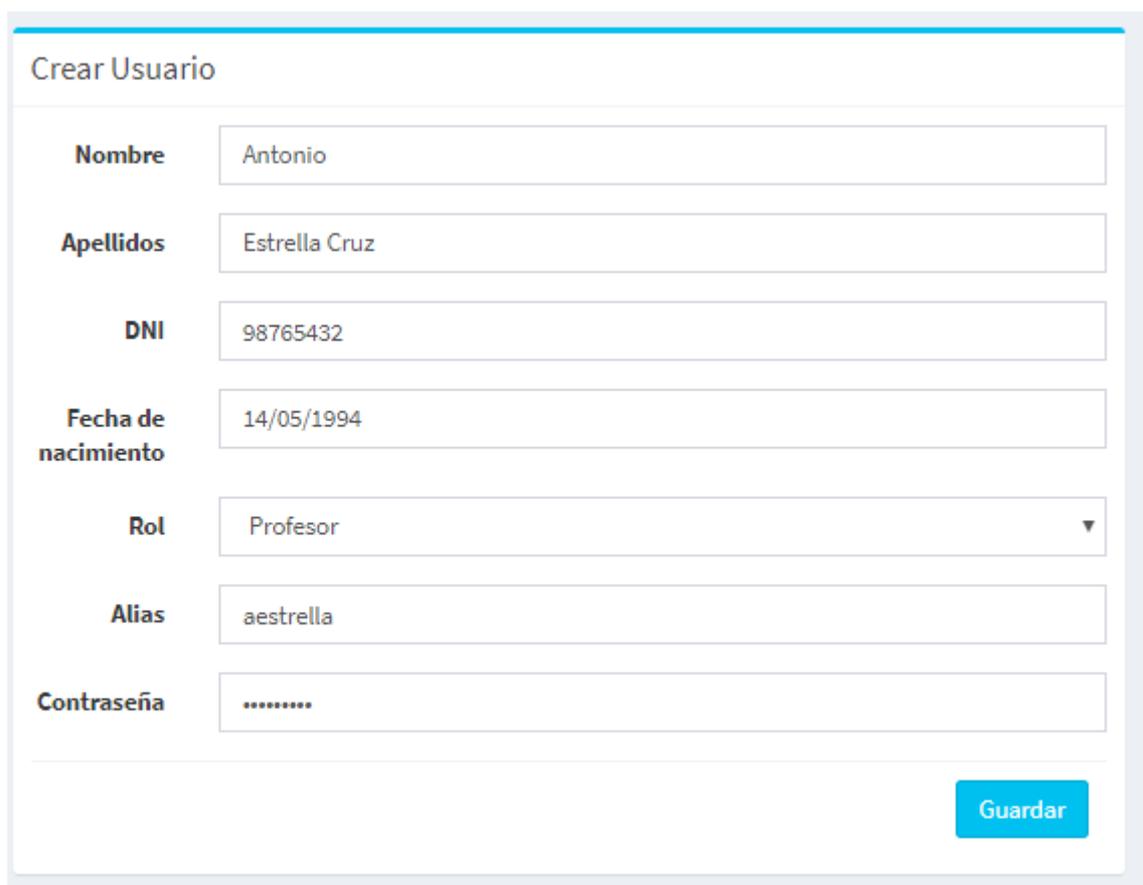
Ilustración 61. Consultar insignias de tipo positivo

Crear usuarios

Esta opción es única y exclusiva del administrador, el cual accederá a ella una vez iniciada la sesión con su perfil de administrador y haciendo click sobre la opción del menú, crear usuarios.

Una vez pulsada esta opción, aparecerá un formulario en el cual deberá introducir todos los datos necesarios para la creación de un nuevo usuario. Ilustración

62



Crear Usuario

Nombre	Antonio
Apellidos	Estrella Cruz
DNI	98765432
Fecha de nacimiento	14/05/1994
Rol	Profesor ▼
Alias	aestrella
Contraseña	*****

Guardar

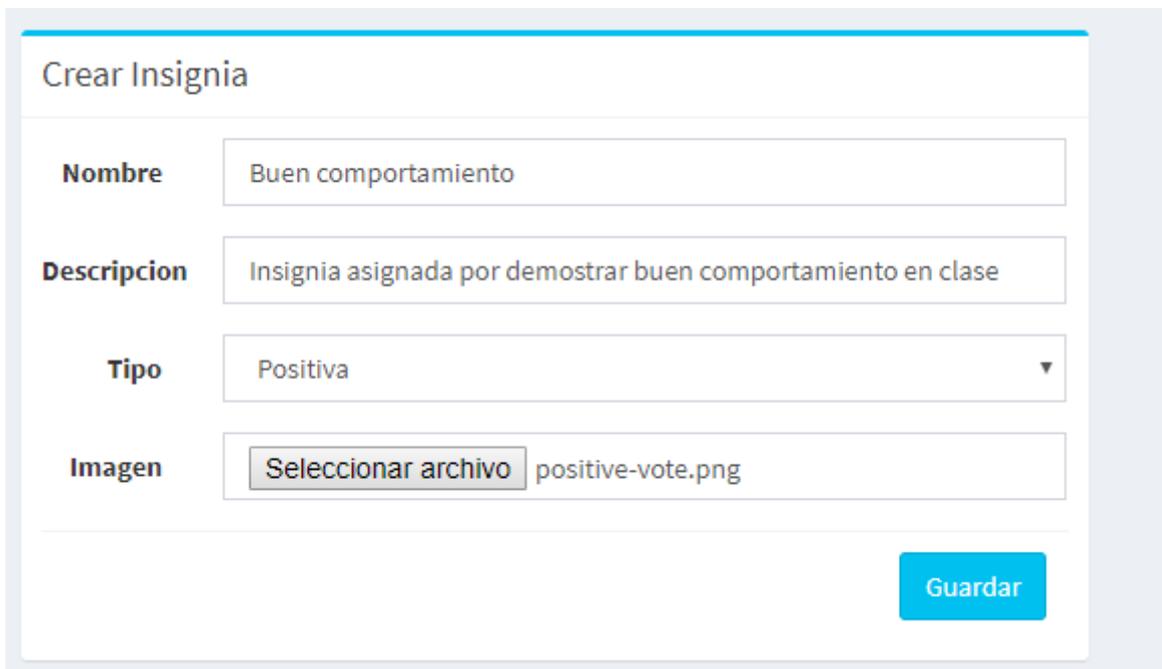
Ilustración 62. Crear usuarios

Crear Insignias

Opción exclusiva de los profesores y del administrador, es muy similar a la creación de usuario.

En primer lugar, se hace click sobre la opción crear insignias, a raíz de esto aparecerá el formulario de creación de insignias, en el que se introducen todos los datos de la insignia junto a una imagen que tengamos previamente descargada en

nuestro sistema, la imagen se redimensionará automáticamente, por lo que no es necesario realizar ninguna modificación de la misma antes de introducirla.



Crear Insignia

Nombre Buen comportamiento

Descripción Insignia asignada por demostrar buen comportamiento en clase

Tipo Positiva ▼

Imagen Seleccionar archivo positive-vote.png

Guardar

Ilustración 63. Creación de Insignias.

Asignar Insignias

Esta opción es exclusiva del profesor, y la realizará para cada alumno que tenga asignado.

El profesor pulsa sobre la opción de ver alumnos que aparece en el menú, y a continuación se le muestra una tabla con el listado de los alumnos y sus datos, a continuación de los datos de los alumnos aparece un botón por cada uno de ellos, que se llama “Asignar Insignias”. Al pulsar ese botón aparecerá una ventana emergente y un desplegable en el que puede elegir qué insignia quiere asignarle al alumno en cuestión.



Listado de Alumnos de Antonio Estrella Cruz

Show 10 entries Search:

ID Alumno	Nombre	Apellidos	Alias	Fecha de nacimiento	Asignar Insignias
16	Antonio	Herrera Melero	adiaz	1995-07-20	Asignar Insignias

Showing 1 to 1 of 1 entries Previous 1 Next

Mostrar

Ilustración 64. Asignación de Insignias



Ilustración 65. Ventana emergente asignación de insignias.

En la Ilustración 65 se muestra la ventana que aparece, el profesor debe seleccionar la insignia de las que aparecen y pulsar sobre el botón de asignar.

Relacionar Profesores con Alumnos

El administrador es el encargado de realizar esta relación mediante la opción que aparece en el menú de asignar profesores. Al hacer click sobre ella, aparece una tabla con un listado de alumnos, en esta tabla aparece al lado de los datos de los alumnos un botón llamado "Asignar Profesor". Una vez pulsado el botón, aparecerá una ventana emergente con los datos del alumno seleccionado, y un desplegable con los distintos profesores del centro.

El siguiente paso es seleccionar el profesor y asignarlo a cada alumno correspondiente.

ID Alumno	Nombre	Apellidos	Alias	Fecha de nacimiento	Asignar Profesores
16	Antonio	Herrera Melero	adiaz	1995-07-20	 Asignar profesor

Ilustración 66. Tabla de asignación de profesores

Asignación de Profesores



Antonio Herrera Melero

Seleccionar profesor

Ilustración 67. Ventana emergente asignación de profesores.

El administrador, solamente debe seleccionar el profesor, y pulsar el botón asignar para que el alumno tenga un profesor asignado.

Relacionar Tutores legales con Alumnos

El proceso es el mismo que el de relacionar profesores con alumnos, solo que esta vez lo realizaría el profesor al pulsar sobre la opción de menú Asignar Tutores legales.

BIBLIOGRAFÍA

- Huotari, K., & Hamari, J. (2012). *Defining gamification*, 17. <https://doi.org/10.1145/2393132.2393137>.
- Doumanis, I., Economoub, D., Robert, G.S. & Porterc, S. (2019). *The impact of multimodal collaborative virtual environments on learning: A gamified online debate*
- Nielsen, J. (2000). *Usabilidad. Diseño de sitios Web*. Prentice Hall.
- Pfleeger, S. L. (s.f.). (2002) *Ingeniería de software, teoría y practica*. Prentice Hall.
- CodeIgniter. (2019). *CodeIgniter.com*. Obtenido de https://www.codeigniter.com/user_guide/
- Zend. (2019). Obtenido de <https://www.zend.com/>
- Phalcon. (2019). Obtenido de <https://phalcon.io/en-us>
- CakePHP. (2019). Obtenido de <https://cakephp.org/>
- PHP. (2019). Obtenido de <https://www.php.net/manual/es/intro-what-is.php>
- JS. (2019). Obtenido de <https://www.w3schools.com/js/default.asp>
- CSS. (2019). Obtenido de <https://www.w3schools.com/css/default.asp>
- HTML. (2019). Obtenido de <https://www.w3schools.com/html/default.asp>
- XAMPP. (2019). Obtenido de <https://www.apachefriends.org/es/index.html>
- Bootstrap. (2019). Obtenido de <https://getbootstrap.com/>
- VisualParadigm. (2019). Obtenido de <https://www.visual-paradigm.com/>
- GanttProject. (2019). Obtenido de <https://www.ganttproject.biz/>
- StackOverflow. (2019). Obtenido de <https://es.stackoverflow.com>