



UNIVERSIDAD DE JAÉN
Nombre del Centro

Trabajo Fin de Grado

APLICACIÓN PARA EVALUAR EN EDUCACIÓN PRIMARIA POR MATERIAS Y COMPETENCIAS

Alumno: Luis Alberto Ortega Ortega

Tutor: Prof. D. Luis Alfonso Ureña López
Don Eleuterio Agea Rodríguez

Dpto: Informática

SEPTIEMBRE, 2016



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Luis Alfonso Ureña López y Don Eleuterio Agea Rodríguez, tutores del Proyecto Fin de Carrera titulado: Aplicación para evaluar en Educación Primaria por materias y competencias, que presenta Luis Alberto Ortega Ortega, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2016

El alumno:

Los tutores:

UREÑA LOPEZ
LUIS ALFONSO
- 52552054K

Firmado digitalmente por UREÑA LOPEZ LUIS ALFONSO - 52552054K
recibido en: (DN) c=ES,
serialNumber=22522046, ou=LOPEÑA LOPEZ,
ou=UNIVERSIDAD DE JAÉN,
LOPEZ LUIS ALFONSO - 52552054K
Fecha: 2016.09.01 12:46:38 +02'00'

AGEA
RODRIGUEZ
ELEUTERIO
- 26487138Q

Firmado digitalmente por AGEA
RODRIGUEZ
ELEUTERIO -
26487138Q Fecha:
2016.09.01 18:52:07
+02'00'

Luis Alberto Ortega Ortega

Luis Alfonso Ureña López
Eleuterio Agea Rodríguez

Índice

1. Introducción	5
1.1. Motivación	5
1.2. Objetivos	5
1.3. Contexto y Competencias	6
1.3.1. CCL	7
1.3.2. CMT	7
1.3.3. CAA	7
1.3.4. CSYC	8
1.3.5. CD	8
1.3.6. CEC	8
1.3.7. SIEP	8
1.4. Estudio Solución Existente	9
1.4.1. Séneca	9
1.5. Estructura del documento	11
2. Análisis	12
2.1. Planificación temporal	12
2.2. Análisis de costes	13
2.2.1. Coste desarrollo	14
2.2.2. Coste funcionamiento	15
2.2.3. Calculo de costes	15
2.3. Especificación Requerimientos	16
2.3.1. Funcionales	16
2.3.1.1. Gestión Usuario	17
2.3.1.2. Gestión Alumnos	18
2.3.2. No funcionales	20
2.4. Casos de Uso	22
2.4.1. Registro	22
2.4.2. Inicio Sesión	23
2.4.3. Perfil	23
2.4.4. Insertar Alumno	24
2.4.5. Consultar Alumno	24
2.4.6. Borrar Alumno	24
2.4.7. Asignatura	25
2.4.8. Evaluar	25
2.4.9. Desconectar	25
2.5. Diagramas de Secuencia	25
2.5.1. Registro	26
2.5.2. Inicio	26
2.5.3. Perfil	27
2.5.4. Borrar usuario	27
2.5.5. Insertar alumno	28
2.5.6. Consultar alumno	29
2.5.7. Borrar alumno	29
2.5.8. Asignatura	30

2.5.9. Evaluar.....	31
2.5.10. Desconectar.....	32
3. Propuesta de Solución.....	34
3.1. Arquitectura.....	34
3.2. Diseño de datos.....	37
3.3. ER.....	38
3.4. Interfaz.....	38
4. Implementación.....	50
4.1. Tecnologías.....	50
4.1.1. WampServer.....	50
4.1.2. Codeigniter.....	50
4.1.3. CSS.....	51
4.2. Herramientas.....	52
4.2.1. NetBeans.....	52
4.2.2. Gliffy.....	52
4.2.3. Microsoft Project.....	52
4.2.4. Marvel App.....	53
4.3. Implementación detallada.....	53
4.3.1. Servidor.....	55
4.3.2. Base de Datos.....	55
4.3.3. Controlador.....	57
4.3.4. Vista.....	58
4.3.5. CSS.....	59
4.3.6. JavaScript.....	60
4.3.7. Evaluación.....	62
5. Evaluación y pruebas de la Aplicación	64
5.1. Registro.....	65
5.2. Inicio.....	65
5.3. Cerrar Sesión.....	65
5.4. Perfil.....	66
5.5. Editar perfil.....	66
5.6. Borrar usuario.....	66
5.7. Ver alumnos.....	67
5.8. Consultar alumnos.....	67
5.9. Introducir alumnos.....	67
5.10. Borrar alumno.....	68
5.11. Asignatura.....	68
5.12. Evaluar.....	69
6. Conclusiones y valoración Personal.....	70
7. Bibliografía.....	73
Anexo I. Manual de Instalación.....	74
Anexo II. Manual de Usuario.....	82
Anexo III. Índice de ilustraciones.....	93

1. INTRODUCCIÓN

En este capítulo se presenta mi Trabajo Fin de Grado. Se trata de una aplicación web para la evaluación de Educación Primaria por materias y por competencias clave a partir de los criterios de evaluación, para saber si un alumno ha cumplido los objetivos y logros durante cada ciclo de educación primaria.

1.1. Motivación del proyecto

La motivación de este proyecto es el aprendizaje de la programación web con lenguajes como son PHP, HTML, JavaScript, además de los CSS para dar estilos y formato a la apariencia de la aplicación. También lo es aportar conocimientos que he adquirido durante estos años en la universidad, para desarrollar una aplicación que ayude a los profesores de Educación Primaria a la evaluación de cada uno de sus alumnos de los diferentes cursos, y que esto facilite a los profesores que dan clases a más de un curso tener todo informatizado en cada momento, y tener un control exhaustivo de cada alumno.

1.2. Objetivos

El objetivo principal es desarrollar una aplicación web para la gestión de alumnos de Educación Primaria y la evaluación de cada uno de ellos tanto por materia como por competencias, una forma de evaluación que está siendo sustituida por la evaluación tradicional. Todo lo referente al alumnado y profesorado se guardará debidamente en la base de datos de PHPMyAdmin.

Los objetivos que debe cumplir la aplicación son:

El usuario (profesor) podrá añadir alumnos por cursos.

El usuario podrá editar sus datos personales.

El usuario podrá evaluar a cada alumno individualmente, por materia y competencias.

1.3. Definición del contexto

Tradicionalmente los profesores de Educación Primaria solían llevar las calificaciones de todos sus alumnos de cada curso apuntado en las llamadas hojas de calificaciones. Esto suponía que los profesores, ante cualquier descuido, pudieran perder las calificaciones de cualquiera de sus alumnos antes de finalizar el curso. También la manera de calificar de los profesores era distinta a la que hay implementada actualmente. Por ejemplo, las notas se dividían en Necesita Mejorar y Progresa adecuadamente.

Con la creación de esta aplicación se pretende evitar los problemas comentados anteriormente, ayudando a los profesores a llevar un control de todos los alumnos desde cualquier dispositivo con conexión a Internet.

También permitirá la evaluación por competencias, además de la tradicional evaluación por materias. La evaluación por competencias pretende poner el acento en aquellos aprendizajes que se consideran imprescindibles, desde un planteamiento integrador y orientado a la aplicación de los saberes adquiridos. De ahí su carácter básico.

La inclusión de las competencias básicas en el currículo tiene varias finalidades. Primero de todo, integrar los distintos aprendizajes, tanto formales, incorporados en cada área o materia, como los informales y no formales. En segundo lugar, permitir a todos los estudiantes integrar sus aprendizajes, ponerlos en relación con distintos tipos de contenidos y utilizarlos de manera efectiva cuando le resulte necesario en diferentes situaciones y contextos. Y, por último, orientar la enseñanza, al permitir identificar los contenidos y los criterios de evaluación que tienen carácter imprescindible y, en general, inspirar las distintas decisiones relativas al proceso de enseñanza y aprendizaje.

Con las áreas y materias del currículo se pretende que todos los alumnos y alumnas alcancen los objetivos educativos y, consecuentemente, también adquieran las competencias básicas. Sin embargo, no existe una relación unívoca entre la enseñanza de determinadas áreas o materias y el desarrollo de ciertas competencias. Cada una de las áreas contribuye al desarrollo de

diferentes competencias y, a su vez, cada una de las competencias básicas se alcanzará como consecuencia del trabajo en varias áreas o materias.

Las competencias básicas para la evaluación en Educación Primaria son:

1. Competencia en Comunicación Lingüística.
2. Competencia Matemática.
3. Competencia para aprender a aprender.
4. Competencia social y ciudadana.
5. Competencia Digital
6. Conciencia y Expresiones culturales.
7. Sentido de Iniciativa y Sentido Emprendedor.

Veamos un poco de cada una de estas competencias básicas.

1.3.1. Competencia en Comunicación Lingüística. (CCL)

Esta competencia se refiere a la utilización del lenguaje como instrumento de comunicación oral y escrita, de representación, interpretación y comprensión de la realidad, de construcción y comunicación del conocimiento y de organización y autorregulación del pensamiento, las emociones y la conducta.

1.3.2. Competencia Matemática. (CMT)

Consiste en la habilidad para utilizar y relacionar los números, sus operaciones básicas, los símbolos y las formas de expresión y razonamiento matemático, tanto para producir e interpretar distintos tipos de información, como para ampliar el conocimiento sobre aspectos cuantitativos y espaciales de la realidad, y para resolver problemas relacionados con la vida cotidiana y con el mundo laboral.

1.3.3. Competencia para Aprender a Aprender. (CAA)

Aprender a aprender supone disponer de habilidades para iniciarse en el aprendizaje y ser capaz de continuar aprendiendo de manera cada vez más eficaz y autónoma de acuerdo a los propios objetivos y necesidades.

1.3.4. Competencia Social y Ciudadana. (CSYC)

Esta competencia hace posible comprender la realidad social en que se vive, cooperar, convivir y ejercer la ciudadanía democrática en una sociedad plural, así como comprometerse a contribuir a su mejora. En ella están integrados conocimientos diversos y habilidades complejas que permiten participar, tomar decisiones, elegir cómo comportarse en determinadas situaciones y responsabilizarse de las elecciones y decisiones adoptadas.

1.3.5. Competencia Digital. (CD)

Es aquella que implica el uso creativo, crítico y seguro de las tecnologías de la información y la comunicación para alcanzar los objetivos relacionados con el trabajo, la empleabilidad, el aprendizaje, el uso del tiempo libre, la inclusión y participación en la sociedad.

1.3.6. Conciencia y Expresiones Culturales. (CEC)

Implica conocer, comprender, apreciar y valorar con espíritu crítico, con una actitud abierta y respetuosa, las diferentes manifestaciones culturales y artísticas, utilizarlas como fuente de enriquecimiento y disfrute personal y considerarlas como parte de la riqueza y patrimonio de los pueblos.

1.3.7. Sentido de Iniciativa y Sentido Emprendedor. (SIEP)

Es la competencia que intenta transformar las ideas en actos.

1.4. Estudio de soluciones existentes.

Actualmente existe una aplicación web que funciona de manera similar a la que he desarrollado.

Veamos de qué se trata:

1.4.1. Nuevo sistema de evaluación en Séneca

Apoyados en un programa de consolidación de las competencias básicas como elemento esencial del currículo y con la intención de mejorar los resultados de la población escolar en el aprendizaje, la Consejería de Educación, Cultura y Deporte (en adelante CECD) de la Junta de Andalucía implanta en el sistema de información Séneca un nuevo sistema de evaluación del aprendizaje del alumnado.

Este nuevo sistema de evaluación se establece en nuestra comunidad autónoma como respuesta a los acontecimientos que se están promoviendo en el sector de la educación, tanto a nivel nacional (COMBAS) como a nivel autonómico (PICBA).

El Proyecto COMBAS, que se desarrolla a nivel estatal, fue la primera acción para crear las condiciones favorables de enseñanza y aprendizaje de las competencias básicas.

Más que valorar la profundidad de los conocimientos que posee el alumnado, la evaluación basada en competencias básicas pretende determinar el nivel de desarrollo conseguido respecto a la capacidad que tienen las personas para seleccionar y aplicar con solvencia los aprendizajes adquiridos en diferentes situaciones y contextos.

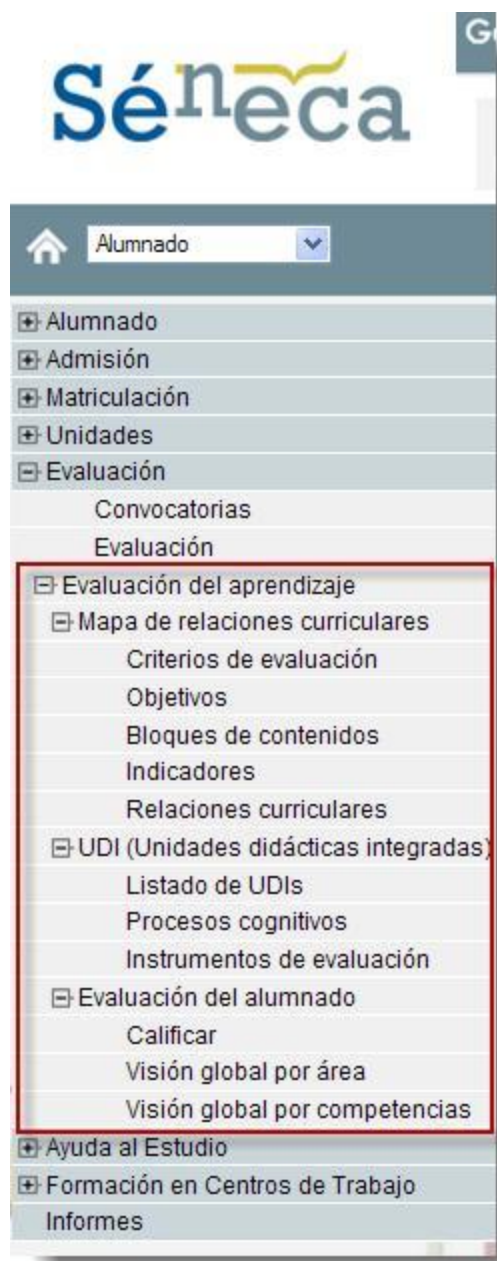


Ilustración 1.1. Séneca

1.5. Estructura del documento.

Vamos a ver la estructura en la que está organizada la documentación del Trabajo Fin de Grado, viendo un resumen del contenido de cada apartado:

- **Introducción:** Se ve una breve descripción del TFG, la motivación para realizarlo, los objetivos a cumplir y alguna solución existente similar a la realizada.
- **Análisis:** aquí vamos a ver las distintas funciones que debe realizar la aplicación ante cualquier petición del usuario.
- **Propuesta de solución:** aquí se define la estructura de la aplicación Web.
- **Implementación:** como se ha desarrollado la aplicación Web, que hemos usado para llevarla a cabo y que herramientas nos han facilitado eso.
- **Evaluación y pruebas:** pruebas realizadas para ver posibles fallos en la aplicación.
- **Conclusión:** Incluye conclusiones obtenidas al realizar el Trabajo Fin de Grado y alguna valoración personal al respecto.
- **Bibliografía:** Material que hemos utilizado para facilitar la realización de éste Trabajo.
- **Anexos:** Aquí incluyo el manual de instalación y manual de usuario para que sea más fácil su ejecución.

2. Análisis

2.1. Planificación del TFG.

Para la realización de cualquier proyecto software, antes de nada, se debe de realizar un planificador de cómo se va a desarrollar el proyecto, indicando el tiempo estimado que va a durar cada parte del mismo.

Es muy difícil que el tiempo final coincida con el estimado, pero hay que acercarse lo más posible.

Vamos a ver cómo queda el TFG dividido en tareas:

Nombre de tarea	Comienzo	Fin
TFG	lun 07/03/16	mié 31/08/16
Bibliografía	lun 07/03/16	jue 14/04/16
Manuales	lun 07/03/16	vie 11/03/16
Tutoriales	lun 14/03/16	jue 14/04/16
Análisis	jue 14/04/16	lun 18/04/16
Requerimiento:	lun 14/03/16	mié 16/03/16
Casos de uso	vie 15/04/16	lun 18/04/16
Modelo-Diseño	mar 19/04/16	mar 26/04/16
Diagrama Secuencia	jue 17/03/16	vie 18/03/16
Modelo ER	mar 19/04/16	jue 21/04/16
Diseño Interfaz	mié 20/04/16	mar 26/04/16
Implementación	mar 26/04/16	jue 25/08/16
Cliente-Servido	mar 26/04/16	jue 28/04/16
Base de datos	vie 29/04/16	mar 03/05/16
Código	mié 04/05/16	vie 19/08/16
Manual de usuario	lun 22/08/16	jue 25/08/16
Pruebas Funcionamiento	vie 26/08/16	mar 30/08/16
Memoria	mar 23/08/16	mié 31/08/16
Completar documentación	mar 23/08/16	mié 31/08/16

Ilustración 2.1. Planificación tareas

Diagrama de Gantt asociado:

Con este diagrama vemos las tareas divididas en el tiempo estimado de ejecución.

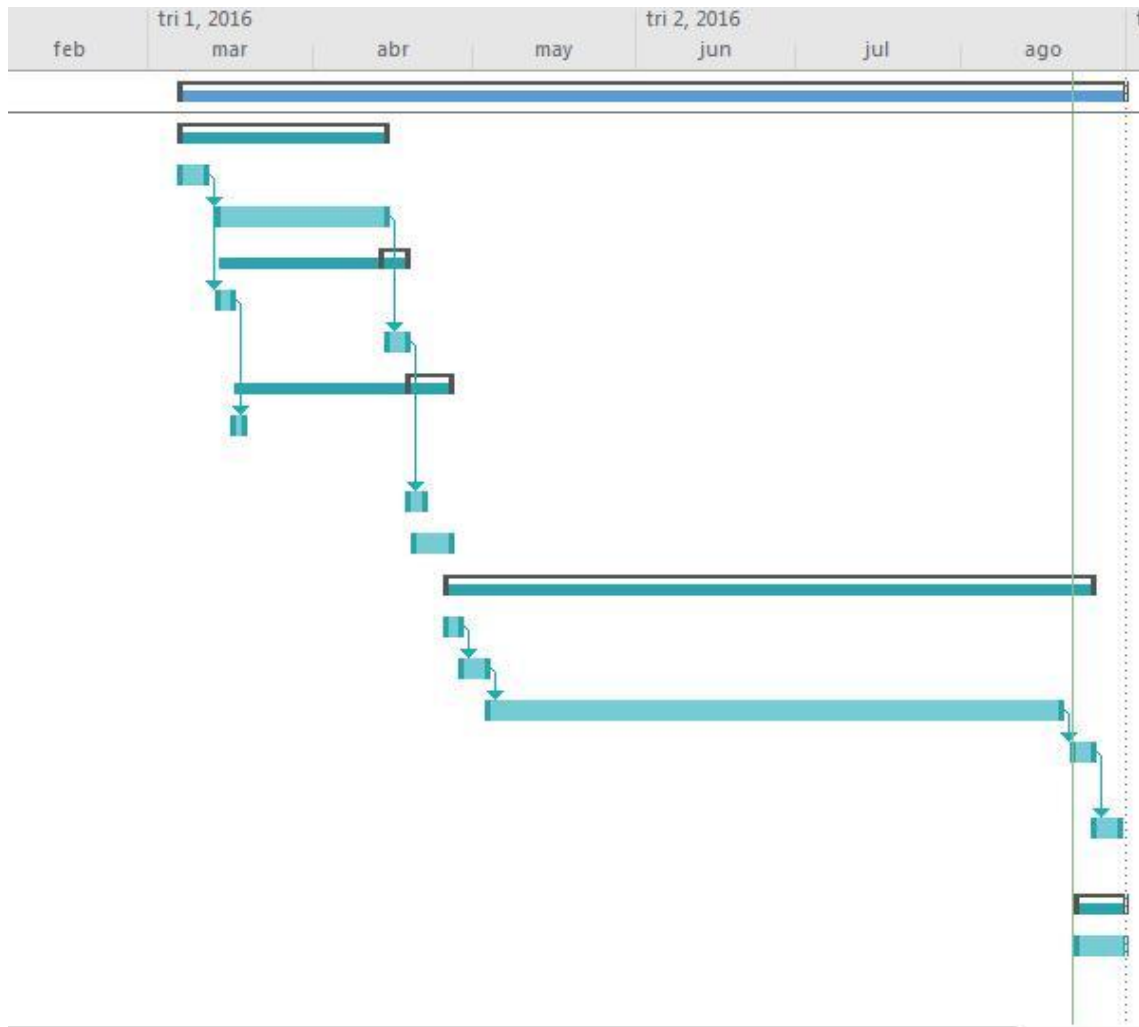


Ilustración 2.2. Diagrama de Gantt.

2.2. Análisis de costes

Aquí vamos a realizar un análisis y cálculo de los costes que conlleva realizar ésta aplicación web y hacerla funcionar.

2.2.1. Coste de desarrollo

Los costes de desarrollo varían en función de la parte en la que tienen lugar. Estos costes solo ocurren una vez durante que se desarrolla la aplicación y no vuelven a producirse una vez que hemos completado por completo la aplicación web.

Se pueden dividir en:

Costes del personal: Sueldo de analistas, programadores, consultores y demás personas que trabajan en el desarrollo de la aplicación. Como es posible que estos trabajadores también lo sean de otro proyecto, habrá que ajustar sus sueldos en función del trabajo estimado sobre nuestro proyecto.

Uso informático: Tiempo que el ordenador trabaja en las diferentes actividades de programación para el desarrollo de la aplicación. Si existiera algún centro informático encargado de aportar sus recursos informáticos para la realización de dicha aplicación también habría que calcular sus costes.

Formación: En caso de que se necesite dar formación a trabajadores para el desarrollo de la aplicación, o en otro caso, los propios usuarios la necesiten para saber el funcionamiento, también supondría gastos.

Costes de suministros y equipos que intervienen en la realización de la aplicación. Así como actualizaciones de software o la compra de nuevos equipos.

2.2.2. Costes de funcionamiento.

Estos costes son asociados una vez se ha puesto en marcha el funcionamiento de la aplicación web.

Mientras que los costes de desarrollo suponían un coste fijo solo una vez durante el desarrollo, este coste se puede producir a lo largo de la vida útil de nuestro sistema, pudiendo ser fijos y variables. Los costes fijos se realizan en intervalos regulares de tiempo y con una cantidad más o menos fija, como son costes de alquiler o licencias software, y también los sueldos de los operadores informáticos y de los trabajadores de soporte de la aplicación. Los variables van asociados según algún factor de utilización, como es el coste de uso informático.

2.2.3. Cálculo de costes.

La aplicación se ha realizado en la plataforma Codeigniter, de distribución gratuita.

Para la realización del Diagrama de Gantt y el planificador de tareas se ha usado Microsoft Project, con licencia académica, pero descargada gratuitamente desde la Universidad.

Para los diagramas de secuencia, también se ha usado plataformas de licencia gratuita.

Para adquirir conocimientos para la formación a la hora de realizar el proyecto, se ha usado manuales y tutoriales de Internet, por lo que no ha supuesto ningún costo.

Por lo tanto, para realizar el cálculo del coste, se tendrá en cuenta el sueldo estimado según los convenios oficiales de los trabajadores informáticos, quedando así: 15 €/hora el analista informático y 10 €/hora el programador.

El horario laboral diario calculado son de 4 horas.

Tarea	Trabajador	Duración(Horas)	Total (€)
Planificación y análisis	Analista	28 horas	420€
Modelo de diseño	Analista	24 horas	360€
Implementación de la aplicación	Programador	352	3520€
Pruebas de funcionamiento	Programador	12 horas	120€
Total			4420 €

Tabla 2.1. Cálculo de costes

2.3. Especificación de requerimientos.

Los requerimientos de una aplicación web son el conjunto de propiedades o restricciones que la aplicación debe de satisfacer.

Estos requerimientos se dividen en dos:

Requerimientos funcionales: se refieren solo y exclusivamente al funcionamiento de la aplicación.

Requerimientos no funcionales: puede referirse a factores externos a la aplicación, pero que de algún modo afecta su rendimiento. Puede ser la velocidad del equipo.

2.3.1. Requerimientos funcionales

Estos requerimientos describen lo que la aplicación debe hacer. Estos requerimientos dependen del tipo de aplicación que se desarrolle y de los usuarios que interactúen con ella.

Los requerimientos funcionales para esta aplicación son los siguientes:

2.3.1.1. Gestión del usuario.

Aquí se lleva a cabo todas las operaciones relacionadas con la gestión del usuario.

Registro de usuario nuevo.

Datos de entrada: Nombre, Apellidos, Correo electrónico, Nombre de Usuario y Contraseña.

Datos de salida: Alerta diciendo que nos hemos registrado satisfactoriamente, o alertas diciendo que algunos campos se encuentran vacíos o mal rellenos.

Una vez que se ha realizado bien el registro, el usuario queda registrado en la base de datos.

2.3.1.1.2. Login al sistema.

Datos de entrada: Nombre de usuario y Contraseña.

Datos de salida: página principal de la aplicación.

Se loguea un usuario en la aplicación. Si los datos son correctos, el sistema lo verifica e inicia sesión. Si no coinciden con los datos de algún usuario de la base de datos, aparece un mensaje de alerta advirtiéndolo que los datos introducidos son incorrectos y vuelve a aparecer las cajas de texto en blanco para introducirlos de nuevo.

Editar perfil del usuario.

Datos de entrada: Nombre, apellidos, contraseña.

Datos de salida: Página de inicio de sesión, una vez modificados los datos.

Desconectar de la aplicación.

Datos de entrada: Pestaña de “Desconectar”.

Datos de salida: Página de inicio de sesión.

Se cerrará la sesión actual y podrá volver a ingresar de nuevo en el sistema o acudir a la página de registro.

Eliminar cuenta de usuario.

Datos de entrada: Botón de eliminar.

Datos de salida: Página de inicio de sesión.

Dentro de la pantalla de Mi perfil aparece un botón para eliminar el usuario. Una vez pulsado dicho botón aparecerá una advertencia para ver si realmente se desea eliminar la cuenta. Si se pulsa aceptar, se saldrá de la aplicación directamente a la página de inicio y el usuario quedará completamente eliminado de la base de datos. En cambio, si se pulsa cancelar, no se eliminará la cuenta.

2.3.1.2. Gestión de los alumnos.

Añadir nuevo alumno.

Datos de entrada: Nombre, Apellidos y Curso del alumno.

Datos de salida: Pestaña para introducir o ver a los alumnos ingresados.

Cuando se añade un alumno, este queda registrado en la base de datos para poder consultarlo y evaluarlo cada vez que se desea.

2.3.1.2.2. Consultar alumnos.

Datos de entrada: Curso.

Datos de salida: Todos los alumnos del profesor actualmente conectado del curso en concreto.

Cuando se pulsa en el botón de consultar, aparecen los nombres y apellidos de cada alumno perteneciente al curso introducido dentro de un botón, desde el que se pulsará para evaluarlos.

2.3.1.2.3. Borrar alumnos

Datos de entrada: Botón borrar.

Datos de salida: Página para volver a consultar o insertar alumnos.

Una vez pulsado el botón borrar que aparece debajo de cada alumno, aparecerá un mensaje de confirmación para borrar o no el alumno. Si se borra, desaparecerá de la base de datos y volverá a la página para seguir consultando. Si se pulsa cancelar, seguirá todo tal y como estaba.

2.3.1.3. Gestión de las evaluaciones.

2.3.1.3.1. Selección del alumno a evaluar

Datos de entrada: Página con el nombre de los alumnos de un curso determinado.

Datos de salida: Página con el nombre de las distintas asignaturas.

Una vez hemos consultado el curso de los alumnos que queremos evaluar, nos aparecerá en pantalla los nombres de los distintos alumnos.

Pulsando encima de uno de ellos nos llevará a la pantalla con el nombre de las asignaturas a evaluar.

2.3.1.3.2. Selección de asignatura.

Datos de entrada: Pestaña con nombre de asignatura.

Datos de salida: Hoja de cálculo de las evaluaciones.

Nos aparecerá una pantalla con el nombre de las diferentes asignaturas.

Cuando pulsemos sobre una de ellas, nos llevará a la pantalla donde se introducen las calificaciones.

2.3.1.3.3. Evaluación del alumno.

Datos de entrada: Valores a introducir en cada una de las casillas de la tabla de evaluación.

Datos de salida: Medias ponderadas, evaluación de las competencias y media de competencias.

Cada vez que introducimos un valor en la tabla y pulsamos la tecla 'Intro' se actualizan los valores de las casillas.

2.3.2. Requerimientos no funcionales.

Son las características o propiedades que de una manera u otra pueden limitar el funcionamiento de la aplicación al usuario.

Estos requerimientos surgen de las necesidades de un usuario para poder interactuar sin ninguna duda con la aplicación en todo momento.

Los requerimientos no funcionales de esta aplicación son:

Usabilidad:

- **Rendimiento:** velocidad de ejecución de las interacciones en la aplicación y los errores.
- **Flexibilidad:** la forma en que la aplicación se adapta a cambios producidos.

Seguridad:

A la aplicación solo pueden acceder los usuarios previamente registrados con éxito y siempre que cumplan con las debidas restricciones a la hora del registro. Un usuario no registrado, nunca podrá hacer uso de la aplicación.

Funcionalidad:

Se realizará comprobaciones de que toda la funcionalidad de la aplicación funciona correctamente para que el usuario, una vez haga uso de dicha aplicación, se adapte lo más rápido posible y haga un uso correcto de la misma.

Documentación:

Todo el desarrollo de la aplicación será documentado convenientemente para facilitar cualquier modificación posterior a su puesta en marcha.

Robustez:

Es la manera en que la aplicación interactúa con el usuario ante posibles fallos, facilitando su uso. También puede ser la capacidad del sistema a tolerar fallos.

Factores relacionados:

- **Navegabilidad:** El usuario debe poder percatarse de donde se encuentra en cada momento sin que le repercuta negativamente.
- **Recuperación información:** La aplicación puede volver a un estado anterior.
- **Tiempo de respuesta:** Tiempo desde que el usuario ejecuta una opción de la aplicación hasta que éste ve los resultados por pantalla.

2.4. Casos de Uso.

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Dicho de otro modo, son las distintas interacciones entre un “actor”, en nuestro caso el usuario, y la aplicación.

En nuestro caso el “actor” sería el usuario (profesor) ya registrado en el sistema.

Veamos las distintas interacciones.

2.4.1. Registro de usuario

1. El usuario que se quiere dar de alta entra a la aplicación.
2. El usuario pulsa Registrarse.
3. El usuario rellena los campos.
 - a. Si existe algún error en los campos, el sistema lo notificará.
4. Sí todos los campos están correctamente rellenos, el sistema notificará al usuario que se ha dado de alta correctamente.
5. Ya tenemos al usuario registrado en la base de datos.

2.4.2. Iniciar sesión.

1. El usuario entra en la aplicación.
2. El usuario introduce su nombre de usuario y contraseña.
 - a. Si los datos son correctos y validados por el sistema, entra en la página principal.
 - b. Si alguno de los datos son incorrectos, se muestran los fallos y se vuelven a introducir los datos.

2.4.3. Modificar perfil.

2.4.3.1. Editar perfil.

1. El usuario pulsa Mi perfil y a continuación Editar.
2. Modifica alguno de los campos que aparecen.
3. Si pulsa en editar, se validan los datos, se guardan y aparece la página de iniciar sesión de nuevo.
4. Si no se desea modificar ningún dato, se pulsa cualquiera de las pestañas que aparecen en la barra de navegación y la aplicación redirigirá al usuario a dicha página.

2.4.3.2. Borrar usuario.

1. El usuario pulsa en la página principal en Mi perfil y a continuación en Borrar.
2. Aparecerá un mensaje para si verdaderamente quieres borrar la cuenta de usuario del sistema.
 - a. Si pulsas Aceptar, se borrará la cuenta, redirigiéndote a la página de Iniciar Sesión.
 - b. Si se pulsa Cancelar, aparecerá la pantalla de Mi Perfil.

2.4.4. Insertar alumno.

1. El usuario pulsa la pestaña de Alumnos.
2. El usuario pulsa sobre Insertar.
3. El usuario rellena los campos.
 - a. Si falta algún campo, el sistema lo notifica.
 - b. Si todos los datos están bien, se añade el alumno.
4. Ya está el alumno dado de alta.

2.4.5. Consultar alumno.

1. El usuario pulsa la pestaña de Alumnos.
2. El usuario introduce en el recuadro de Introducir curso, el curso que desee.
3. El sistema devuelve todos los alumnos dados de alta por dicho usuario y del curso indicado.

2.4.6. Borrar alumno.

1. El usuario consulta el curso al que pertenece el alumno interesado en borrar.
2. El sistema muestra los alumnos de dicho curso.
3. El usuario pulsa sobre borrar el alumno que desee.
4. La aplicación muestra un mensaje de alerta para confirmar el borrado.
 - a. Si se pulsa Aceptar, se borra el alumno.
 - b. Si se pulsa Cancelar, siguen apareciendo los alumnos del curso consultado.
5. Si se ha borrado, aparece de nuevo para consultar los alumnos.

2.4.7. Seleccionar Asignatura.

1. El usuario desde la pantalla de alumnos consultados, selecciona al que desee evaluar.
2. El sistema muestra las asignaturas de dicho curso.
3. El usuario selecciona la asignatura a evaluar.

2.4.8. Evaluar Asignatura.

1. El usuario selecciona el alumno a evaluar.
2. El usuario selecciona la asignatura que desee evaluar.
3. El usuario rellena la tabla de calificaciones.
4. El sistema calcula las notas medias y la rúbrica de cada competencia clave.

2.4.9. Desconexión del sistema.

1. El usuario selecciona la opción desconectar de la página principal.
2. El sistema destruye tu sesión y te redirige a la página de inicio de sesión.

2.5. Diagramas de Secuencia.

Los diagramas de secuencia muestran cómo interactúan el usuario de nuestra aplicación y la propia aplicación en un grafo. El diagrama consta de objetos en el que aparecen mensajes, que son representados por líneas continuas acabadas en flecha entre los objetos que interactúan en la aplicación, y el tiempo es representado por una progresión vertical.

2.5.1. Registrar Usuario

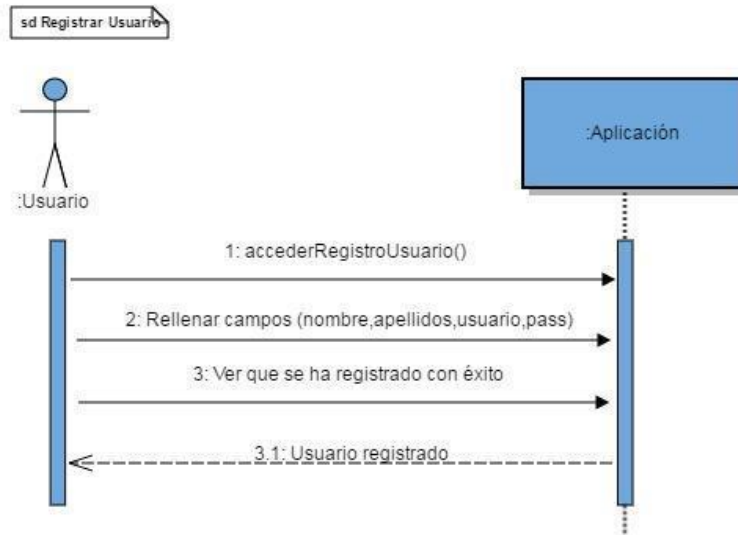


Ilustración 2.3. "Diagrama Registro"

2.5.2. Iniciar Sesión

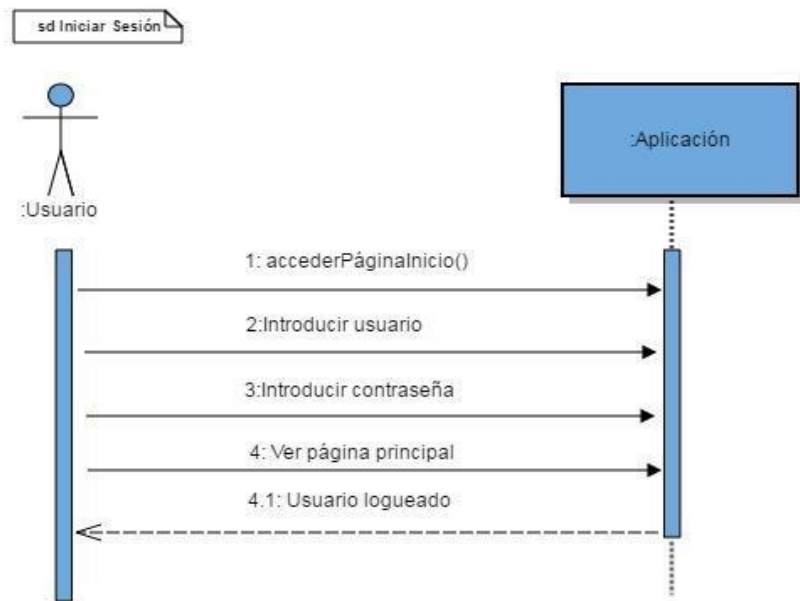


Ilustración 2.4. "Diagrama Inicio Sesión"

2.5.3. Editar Perfil

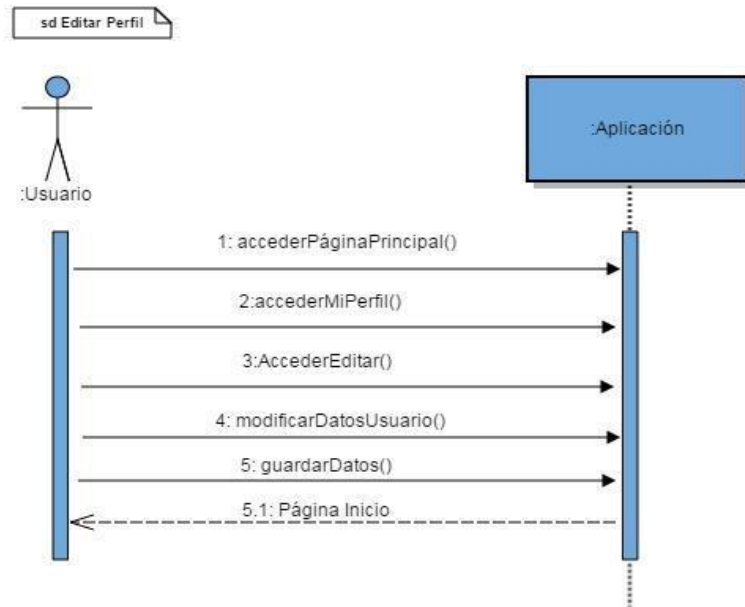


Ilustración 2.5. "Diagrama Editar Perfil"

2.5.4. Borrar Usuario

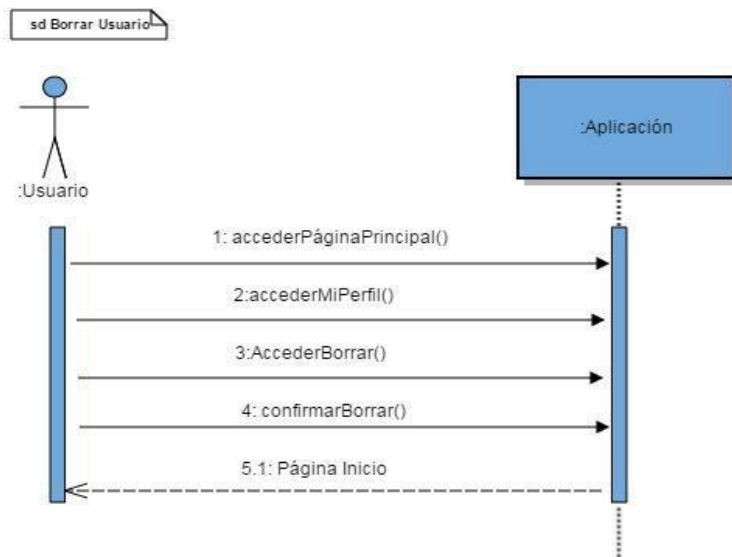


Ilustración 2.6. "Diagrama Borrar Usuario"

2.5.5. Introducir Alumno

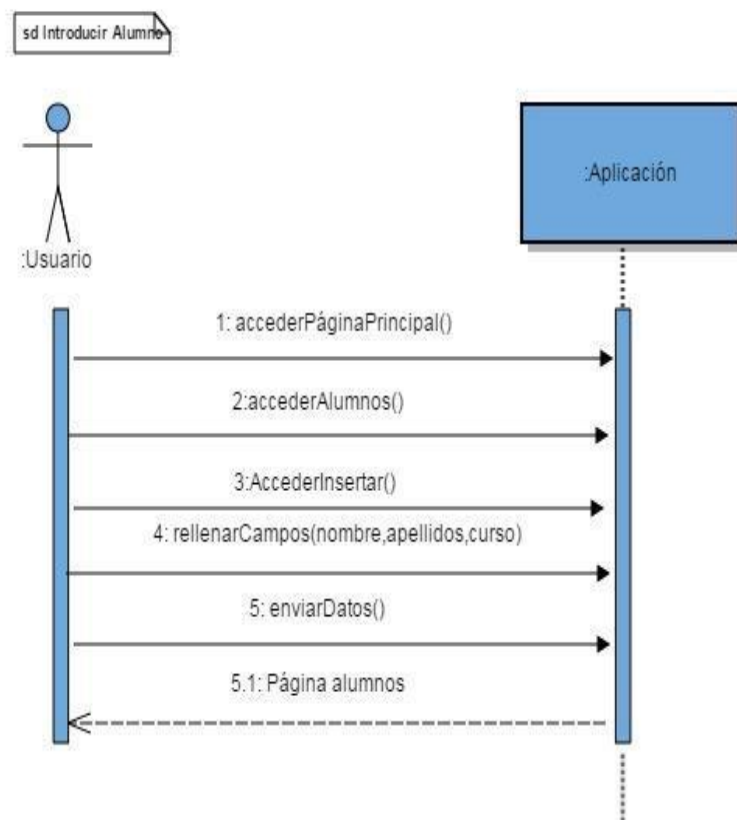


Ilustración 2.7. "Diagrama Insertar Alumno"

2.5.6. Consultar Alumno

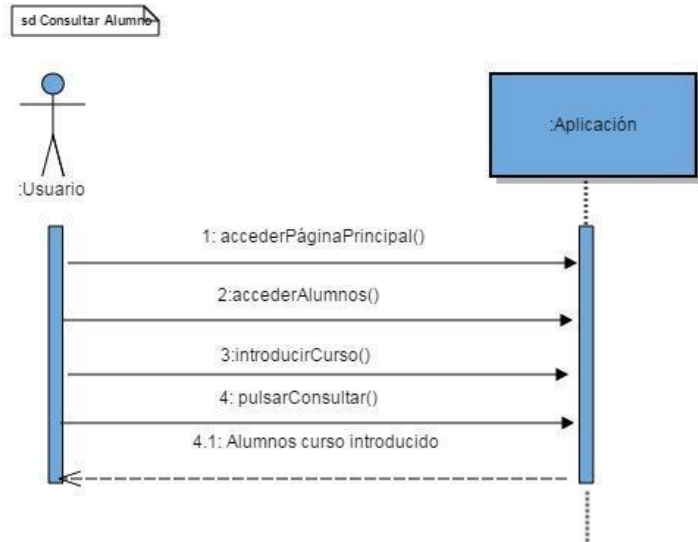


Ilustración 2.8. "Diagrama Consultar alumno"

2.5.7. Borrar Alumno

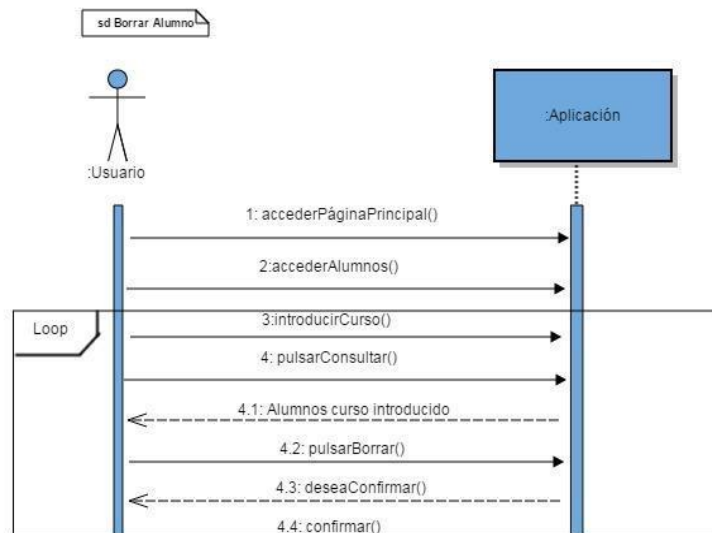


Ilustración 2.9. "Diagrama Borrar Alumno"

2.5.8. Elegir Asignatura

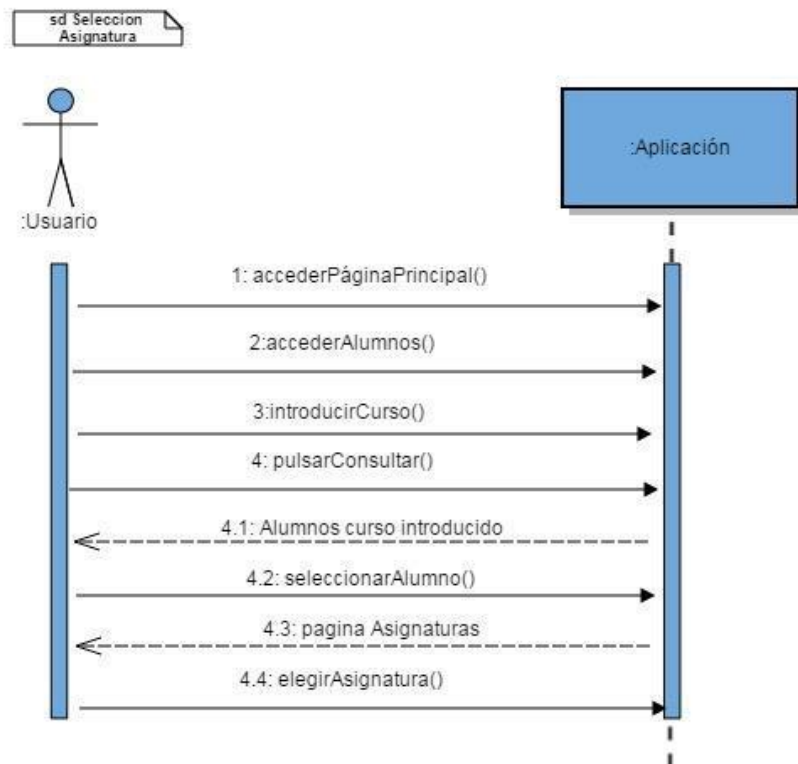


Ilustración 2.10. "Elegir Asignatura"

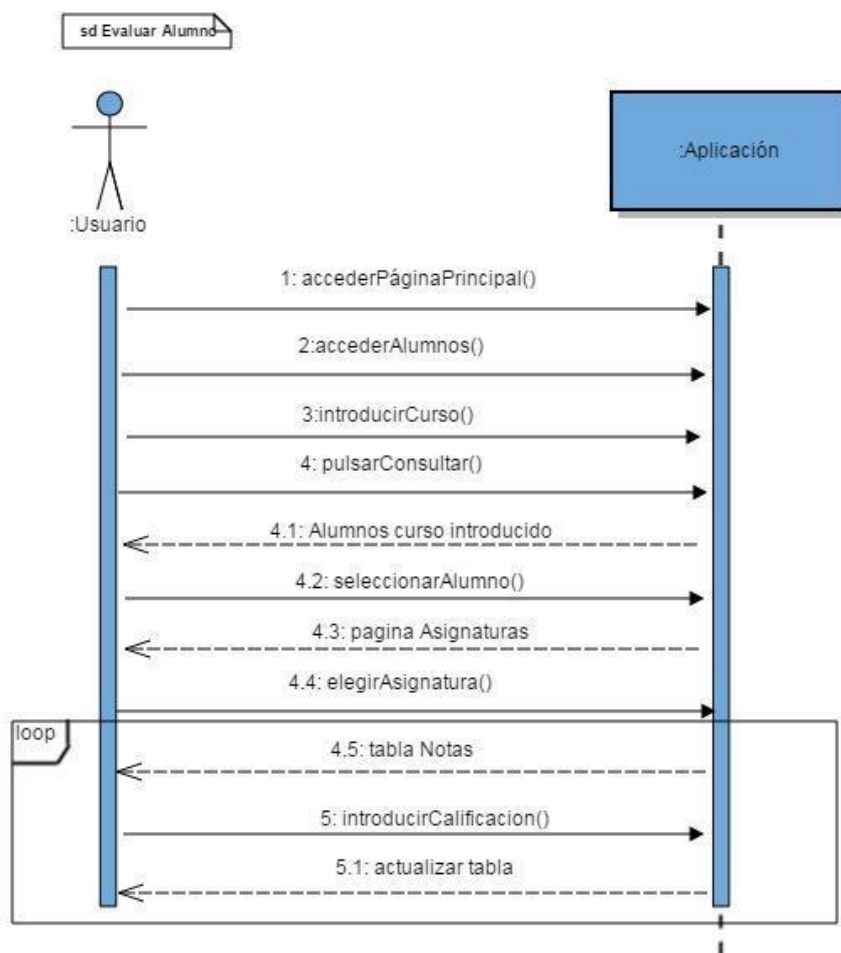
2.5.9. Evaluar Alumno

Ilustración 2.11. “Diagrama Evaluación”

2.5.10. Desconexión de la aplicación.

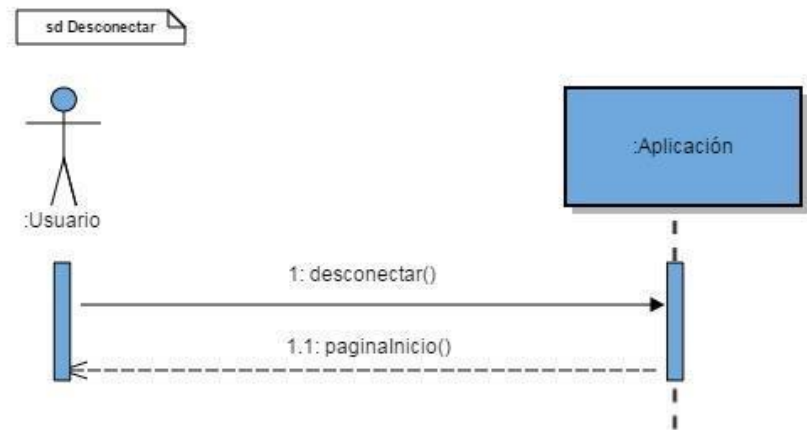


Ilustración 2.12. "Diagrama Desconectar"

3. Propuesta de solución.

3.1. Arquitectura de la solución.

La arquitectura usada para realizar la aplicación ha sido por capas: cliente/servidor, además de la base de datos.

La principal ventaja es que se puede desarrollar por niveles, lo que favorece que cualquier cambio en una capa no afecte a las demás.

Capa Cliente.

Es la capa que ve el usuario, la llamada interfaz gráfica en la que el usuario podrá interactuar con la aplicación, y en la que el sistema le dará información al propio usuario.

Capa Servidor.

Es la que se encarga de procesar las peticiones del usuario. Esta capa al recibir las peticiones, las procesa y envía las respuestas al usuario.

También es la capa que está conectada con la base de datos para almacenar o coger datos de ella.

Base de datos.

En ella se encuentra almacenada toda la información y datos referentes a la aplicación web. Siempre está disponible para almacenar datos o para recuperarlos cuando el servidor se lo pida.

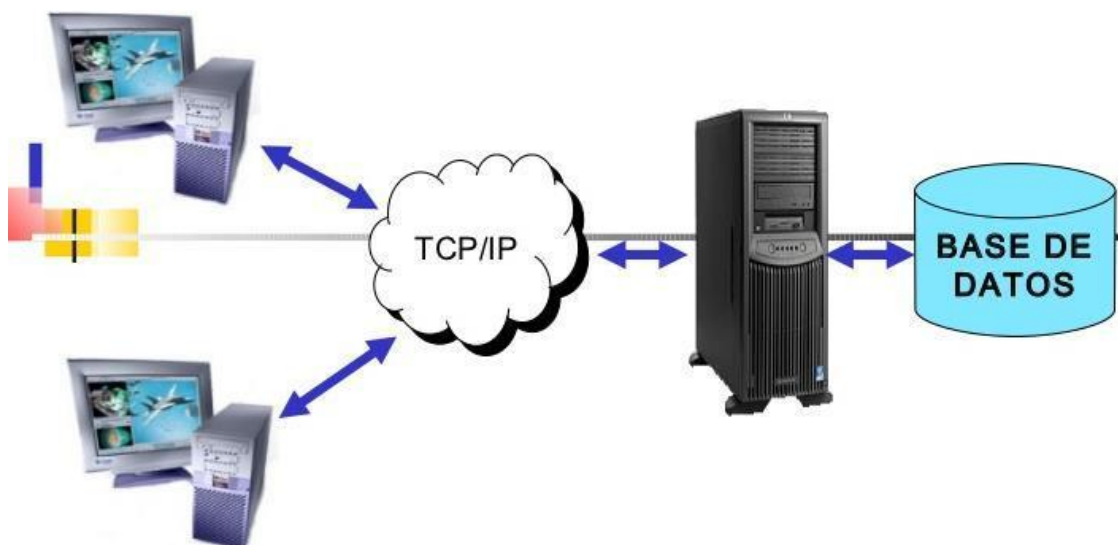


Ilustración 3.1. "Arquitectura"

La aplicación no precisa de conexión a Internet, ya que se ha realizado todas las conexiones de manera local.

En cuanto al patrón de arquitectura utilizado para realizar la aplicación, ha sido el más que conocido **MVC** (Modelo Vista Controlador), utilizado dentro del Framework de Codeigniter.

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

Para ello **MVC** 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.

Veamos más detenidamente cada una de las capas:

Modelo: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información que reside en nuestro caso en la base de datos, tanto consultas como actualizaciones, implementando

también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio).

Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

Vista: Presenta el 'modelo' (información y *lógica de negocio*) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'Modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'Vista' asociada si se solicita un cambio en la forma en que se presenta el 'Modelo', por tanto se podría decir que el 'controlador' hace de intermediario entre la 'Vista' y el 'Modelo'

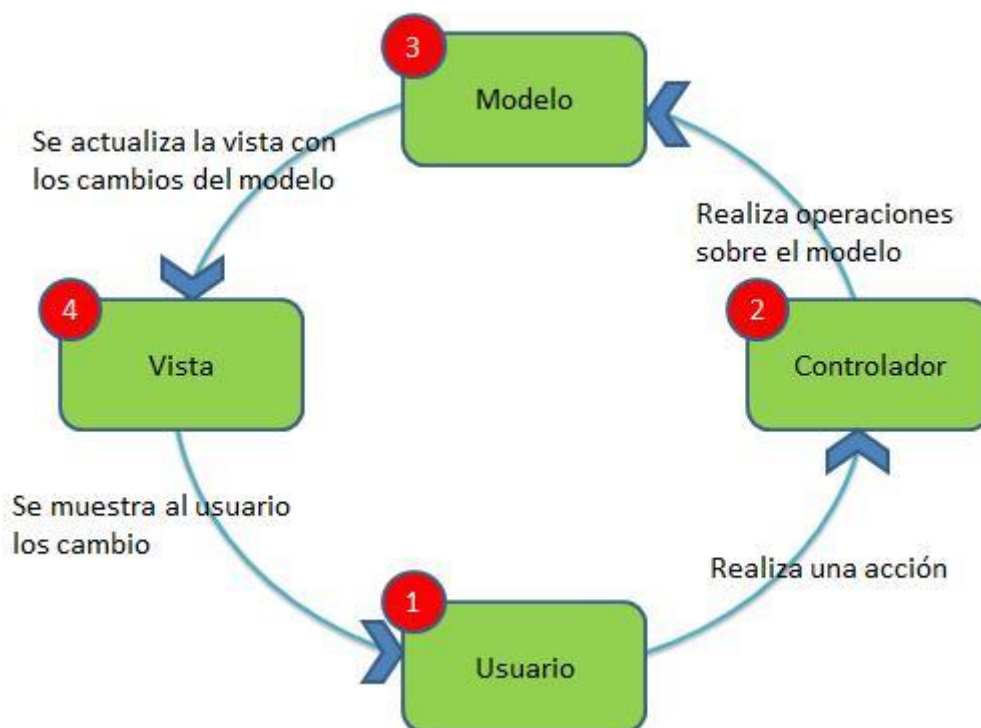


Ilustración 3.2. "MVC"

3.2. Diseño de datos.

El diseño de datos es un proceso que abarca desde la cuestión más elemental, "¿Qué datos requiere la aplicación?", hasta los procesos y estructuras de datos precisos que proporcionan dichos datos. Si el diseño de datos es bueno, el acceso a los datos de la aplicación será rápido y fácil de mantener, y podrá aceptar sin problemas las futuras mejoras de los datos.

Para este sistema se va a diseñar una base de datos para almacenar la información necesaria que de soporte a los futuros accesos a dicha información. La base de datos tendrá las distintas entidades o tablas:

3.2.1. Usuarios.

Id (objectId)

Nombre (String)

Apellidos (String)

Correo (String)

Usuario (String)

Password (String)

3.2.2. Alumnos.

Id (objectId)

Nombre (String)

Apellidos (String)

Curso (Integer)

Profesor (String)

3.3. Modelo Entidad Relación.

Un diagrama o modelo entidad-relación (a veces denominado por sus siglas en inglés, *E-R* "Entity Relationship", o del español *DER* "Diagrama de Entidad Relación") es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.

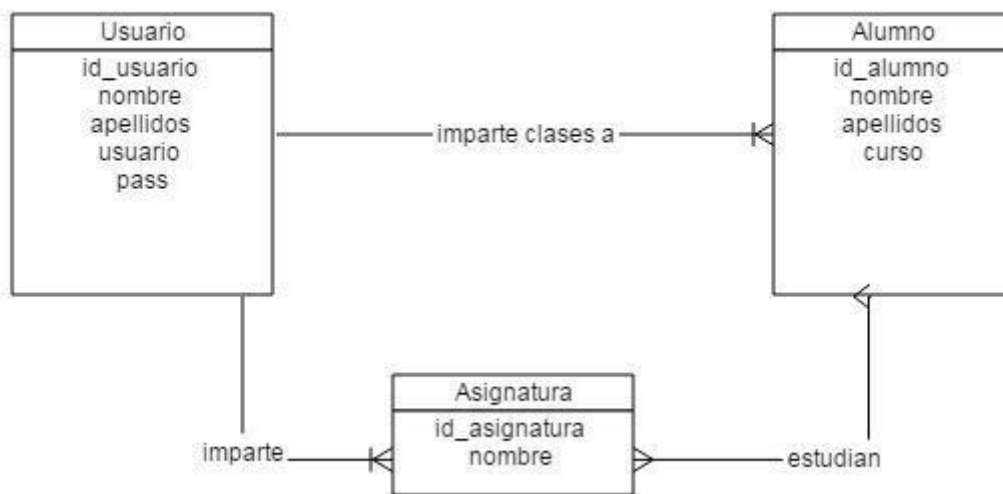


Ilustración 3.3. "Diagrama ER"

3.4. Diseño de la interfaz.

El diseño de interfaz de usuario o ingeniería de la interfaz es el diseño de computadoras, aplicaciones, máquinas, dispositivos de comunicación móvil, aplicaciones de software, y sitios web enfocado en la experiencia de usuario y la interacción.

Su objetivo es que las aplicaciones o los objetos sean más atractivos y además, hacer que la interacción con el usuario sea lo más intuitiva posible, conocido como el diseño centrado en el usuario. En este sentido las disciplinas del diseño

industrial y gráfico se encargan de que la actividad a desarrollar se comunique y aprenda lo más rápidamente, a través de recursos como la gráfica, los pictogramas, los estereotipos y la simbología, todo sin afectar el funcionamiento técnico eficiente.

3.4.1. Diseño centrado en el usuario

El Diseño Centrado en el Usuario es una filosofía de diseño que tiene por objeto la creación de productos que resuelvan necesidades concretas de sus usuarios finales, consiguiendo la mayor satisfacción y mejor experiencia de uso posible con el mínimo esfuerzo de su parte.

Toma forma como un proceso en el que se utilizan una serie de técnicas multidisciplinarias y donde cada decisión tomada debe estar basada en las necesidades, objetivos, expectativas, motivaciones y capacidades de los usuarios.

Hay unas series de principios centro del diseño centrado al usuario que vamos a desgranar:

El **control de la situación** debe de estar en manos del usuario: debe ser el usuario quien inicie las acciones y controle las tareas. La aplicación debe ser lo más interactiva posible, facilitando el cambio y gestión de las funciones.

La **consistencia** es indispensable en el diseño: la aplicación debe ser consistente para que mejore la usabilidad, debe de dar siempre la misma respuesta ante la misma acción del usuario.

No se puede descuidar la **estética**: determinados atributos visuales concentran la atención del usuario en la tarea que están desarrollando.

Hay que posibilitar la resolución ante **errores**: hay que posibilitar la reversibilidad y la recuperabilidad en las acciones.

Se distinguen cuatro partes en el proceso de diseño:

1. Definición clara de los objetivos, entendiendo a los usuarios y contemplando factores como la edad, la experiencia, las limitaciones físicas, las necesidades más especiales, el entorno de trabajo, las influencias sociales y culturales... Hay que definir el marco de trabajo conceptual para presentar el producto en cuestión con el conocimiento y la experiencia de la audiencia objetivo; a partir de ahí, procede una documentación apropiada a este estado.
2. Comunicación del diseño mediante el prototipado y establecimiento de un flujo de tareas. Se puede tratar de incluir más aspectos y comprobar la reacción a los mismos de los usuarios objetivo o tratar de centrarse en los detalles de dichos aspectos, en su funcionalidad.
3. Mediante el test, en el proceso de diseño, la participación del usuario proporciona la inestimable ayuda de determinar en qué medida el producto se está ajustando a las necesidades y a las expectativas creadas. No se trata tanto de evaluar la eficiencia de las tareas y los posibles errores en el diseño, sino de conocer las percepciones del usuario, su satisfacción, sus preguntas, sus problemas,...
4. Después del test va a ser preciso el rediseño en mayor o menor medida, tras el cual inevitablemente, es preciso de nuevo el test, volviendo así a iniciar el ciclo.

Para terminar es importante enumerar las ventajas que aporta para la empresa el DCU:

- Aplicaciones/webs más fáciles de comprender y utilizar y, por lo tanto reducen los costes de formación y asistencia técnica.
- Mejora la eficacia, eficiencia y satisfacción del usuario reduciendo así las molestias y el estrés en cuanto a las aplicaciones/webs
- Mejora la calidad y el atractivo para los usuarios.

Productos/webs más competitivos por lo que obtendremos más ventas y más visitas.

3.4.2. Prototipado.

Los prototipos son visiones esquemáticas de una interfaz donde se sitúan los elementos básicos de una página como el menú de navegación, pestañas de visita a otras pantallas, etc., sin elementos gráficos. También pueden llamarse Wireframes.

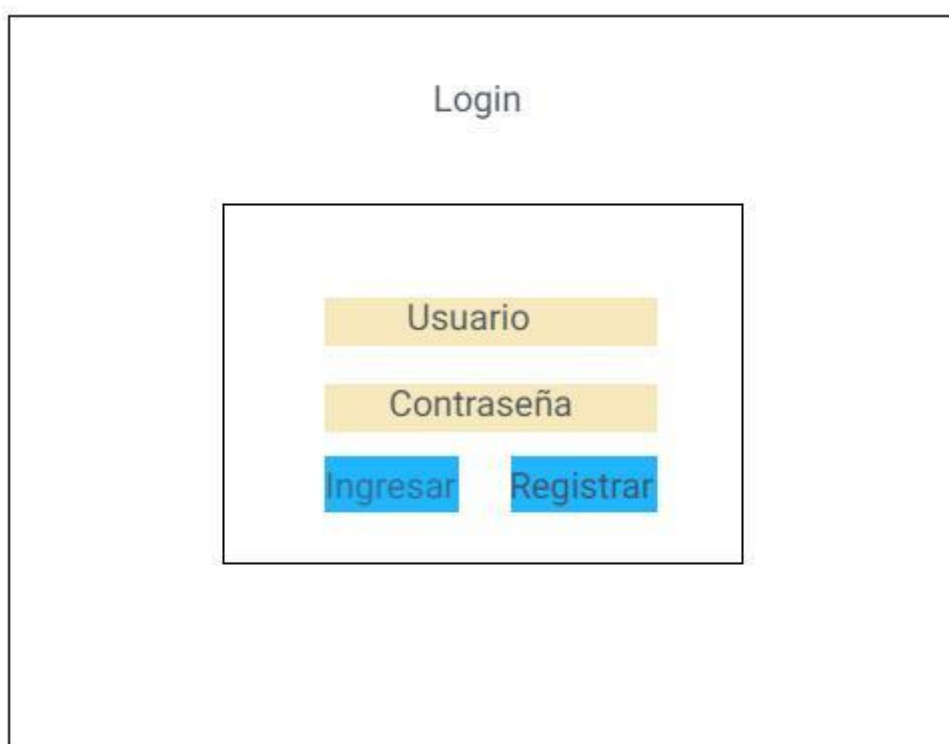
El prototipado puede ir desde un papel donde se visualiza las diferentes interacciones entre la interfaz y el usuario, hasta un completo software.

Estos prototipos ayudan a ahorrar costes y tiempo, ya que en ellos podemos ver las estructuras sin mucho detalle y los clientes pueden dar el visto bueno antes de empezar diseñar y que haya que hacer grandes cambios una vez hayamos empezado la aplicación. También nos ayuda a pensar en cómo se va a comportar la navegación entre páginas, y que va a ver el usuario cuando interactúe con cualquier objeto de la pantalla.

A continuación veremos unos prototipos de cómo quedaría nuestra aplicación web.

3.4.2.1. Inicio de sesión.

Para iniciar sesión ingresando el nombre de usuario y contraseña o para ir a registrar nuevo usuario.



El prototipo muestra una pantalla con el título "Login" centrado en la parte superior. Debajo del título, hay un recuadro que contiene dos campos de entrada de texto, uno etiquetado "Usuario" y otro "Contraseña", ambos con un fondo amarillo. Debajo de estos campos, hay dos botones azules: "Ingresar" a la izquierda y "Registrar" a la derecha.

Ilustración 3.4. "Prototipo Inicio"

3.4.2.2. Registro usuario.

Formulario para rellenar y crear un nuevo usuario, siempre que se completen debidamente los datos. O para volver a la página de Inicio de Sesión.



El prototipo de registro muestra un formulario con el título "Registrar" centrado en la parte superior. El formulario está contenido dentro de un recuadro rectangular con un borde negro. Dentro de este recuadro, hay cuatro campos de entrada de texto, cada uno con un fondo amarillo y un borde negro, que están etiquetados como "Nombre", "Apellidos", "Usuario" y "Contraseña" desde arriba hacia abajo. Debajo de estos campos, hay dos botones rectangulares con fondo azul y texto blanco: "Registrar" a la izquierda y "Login" a la derecha. El recuadro del formulario está rodeado por una línea horizontal superior y una línea vertical a la derecha que se extienden fuera de los límites del recuadro mismo.

Ilustración 3.5. "Prototipo Registro"

3.4.2.3. Página Principal.

En la página principal podemos ver una barra de navegación para acceder a las distintas pantallas de nuestra aplicación. Entre las que vemos están: Home, donde explican de que va la aplicación y autor, Perfil, donde aparecen los datos del usuario que está conectado en el sistema, Alumnos, donde vemos los alumnos de dicho usuario por curso y Desconectar, para salir de la sesión iniciada actualmente.

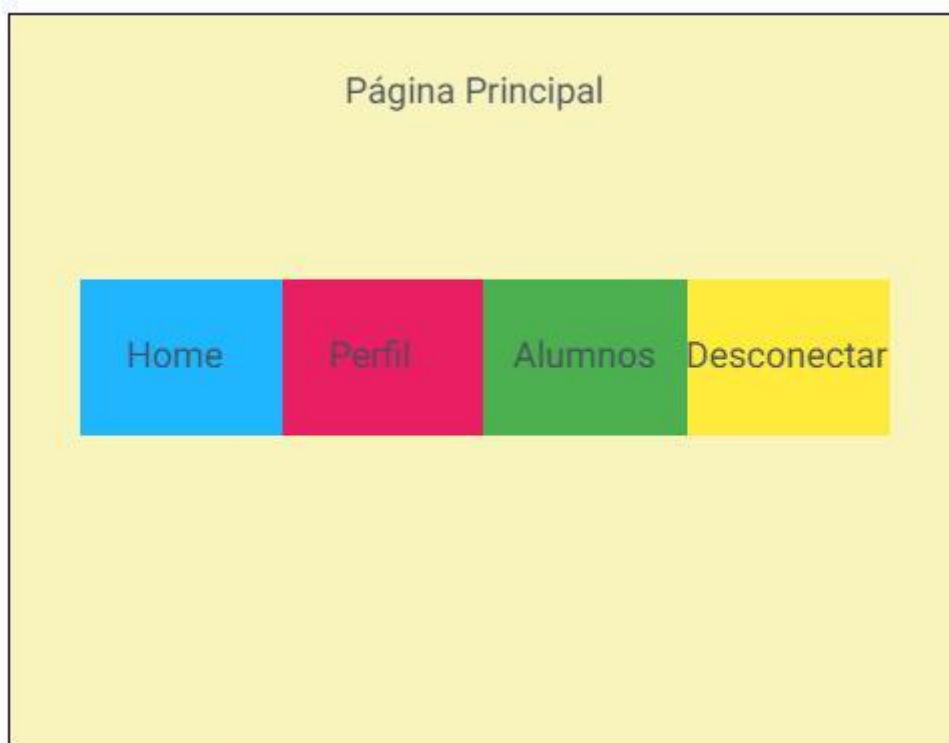


Ilustración 3.6. "Prototipo Principal"

3.4.2.4. Perfil.

En esta sección podemos ver los datos del usuario que ha iniciado sesión, así como las opciones de editar dichos datos y borrar la cuenta del usuario en cuestión.

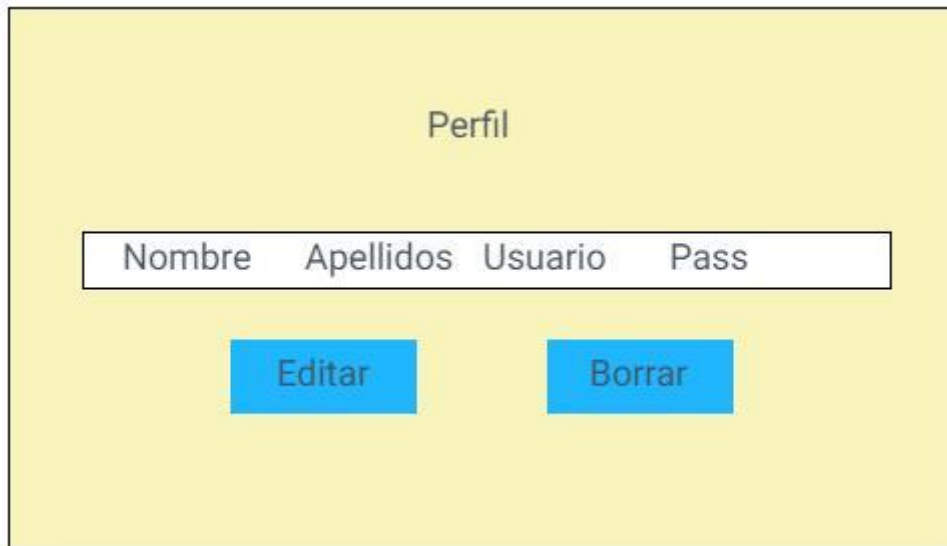


Ilustración 3.7. "Prototipo Perfil"

3.4.2.5. Editar perfil

En esta pestaña, podemos editar algunos datos personales del usuario.

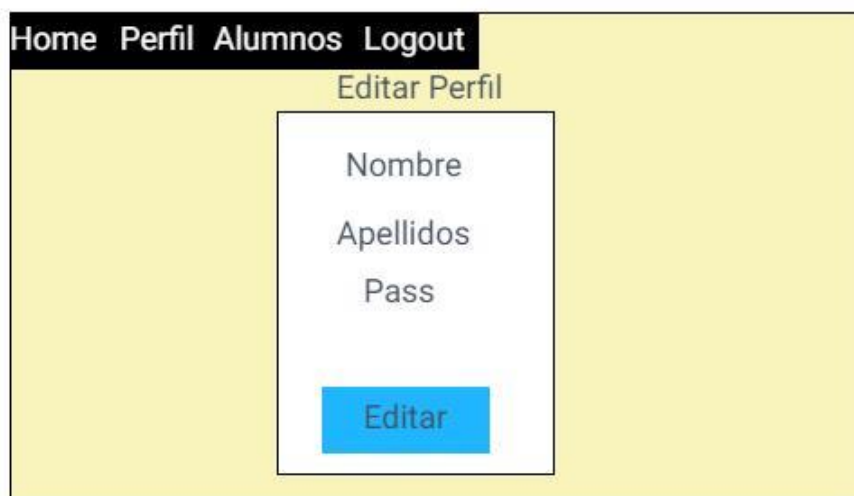


Ilustración 3.8. "Prototipo Editar Perfil"

3.4.2.6. Alumnos.

Aquí podemos introducir un curso para ver los alumnos insertados por dicho usuario que pertenecen a ese curso. También podemos acceder a la página de insertar un nuevo alumno.



Ilustración 3.9. "Prototipo Alumnos"

Alumnos Consultados.

Podemos ver todos los alumnos del curso introducido en la pestaña anterior. Junto al nombre del alumno podemos ver el botón de borrar para eliminar dicho alumno de la base de datos y si pulsamos sobre él accederemos a una pantalla para poder evaluarlo seleccionando una asignatura.

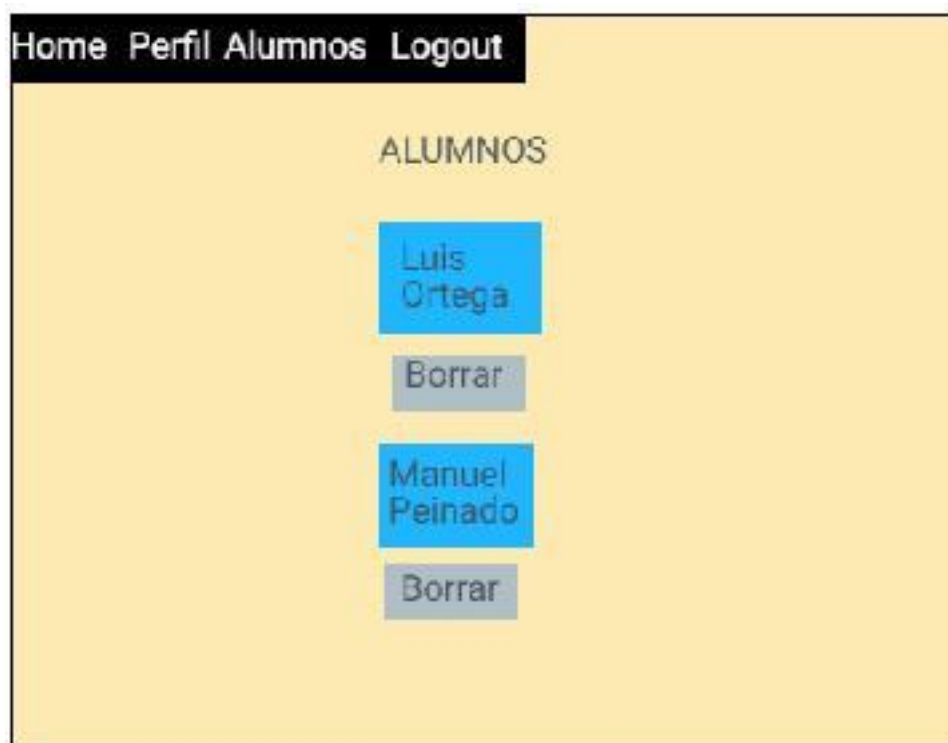


Ilustración 3.10 Prototipo 'Alumnos Consultados'

3.4.2.7. Seleccionar Asignatura.

En esta pantalla, que nos aparece tras pinchar sobre un alumno consultado, nos aparecen las distintas asignaturas a seleccionar para evaluarlas.



Ilustración 3.11 Prototipo 'Asignaturas'

3.4.2.8. Evaluar Asignatura.

Aquí será donde el profesor evaluará las materias de los alumnos, así como podrá ver los logros en cuanto a competencias conseguidos.

LENGUA			
Indicadores	Examen	Libreta	Vocabulario
Indicador 1	8	9	8
Indicador 2	9	8	8
Indicador 3	9	9	8
Indicador 4			9
Indicador 5			

Ilustración 3.12 Prototipo 'Evaluar Asignatura'

4. IMPLEMENTACIÓN

En este capítulo veremos todo lo necesario para llevar a cabo la aplicación, tanto herramientas, tecnologías y otras partes importantes para que todo funcione correctamente.

4.1. Tecnologías.

La aplicación se desarrollará como dijimos anteriormente bajo la arquitectura cliente-servidor y junto a una base de datos para el almacenamiento. En la aplicación hemos usado distintos lenguajes, como son: PHP, que compone mucha parte del código de la aplicación, HTML, que también supone una parte bastante considerable, y también JavaScript para algunas alertas y mensajes.

4.1.1. WampServer.

¿Qué es WampServer? Es un entorno de desarrollo web para Windows con el que se podrán realizar aplicaciones web usando código PHP, y base de datos con MySQL y PHPMyAdmin para manejarla a nuestro antojo.

Provee a los desarrolladores con los cuatro elementos necesarios para una aplicación web: un sistema operativo (Windows), un manejador de base de datos (MySQL), un software para servidor web (Apache) y un software de programación web, en nuestro caso PHP.

4.1.2. Codeigniter.

Probablemente ya sepamos que un framework es un programa para desarrollar otros programas. Codeigniter, por tanto, es un programa o aplicación web desarrollada en PHP para la creación de cualquier tipo de aplicación web bajo PHP. Es un producto de código libre, libre de uso para cualquier aplicación.

Como cualquier otro framework, Codeigniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de

desarrollarlas que debemos seguir para obtener provecho de la aplicación. Esto es, marca una manera específica de codificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado y sea más fácil de crear y mantener. CodeIgniter implementa el proceso de desarrollo llamado Model View Controller (MVC), que es un estándar de programación de aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales, que ya explicamos anteriormente.

Lo que me ha llevado a usar CodeIgniter son varios motivos, entre ellos: conocía este framework de alguna asignatura durante el Grado, es muy fácil de aprender, está perfectamente documentada y además existen muchos tutoriales que te facilitan las cosas.

4.1.3. CSS.

CSS es un lenguaje de estilo que define la presentación de los documentos HTML. Por ejemplo, CSS abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y muchos otros temas.

Es posible usar HTML, o incluso abusar del mismo, para añadir formato a los sitios web. Sin embargo, CSS ofrece más opciones y es más preciso y sofisticado. CSS está soportado por todos los navegadores hoy día. En mi caso CSS me ayuda a añadir estilo a algunas fuentes, pero la mayor parte del código es de HTML.

4.2. Herramientas usadas en el TFG

4.2.1. NetBeans.

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java, pero que cuenta con múltiples módulos extensibles para otros lenguajes de programación, en mi caso PHP, HTML y JavaScript.

Por lo tanto esta herramienta la he usado para generar todo el código de la aplicación, ya que estaban todos los archivos ordenados y mejor localizados que en cualquier otra herramienta.

4.2.2. Gliffy.

Es una herramienta online que nos permite hacer diagramas de casi cualquier tipo. Soporta diagramas como son: UML, diagramas de Redes, diagramas Entidad Relación, Diagramas de caso de Uso...

En mi caso lo he utilizado para el diagrama ER y para los diagramas de Secuencia de las distintas interacciones entre Usuario y Aplicación.

4.2.3. Microsoft Project.

Esta herramienta la he usado para la planificación de las distintas tareas en las que se divide el total del Trabajo Fin de Grado. También una vez creadas las tareas y planificadas según el orden, te aporta el Diagrama Gantt asociado a dichas tareas.

4.2.4. Marvel App.

Es una aplicación que se encuentra tanto Online como en descargas que nos ayuda a realizar sencillos prototipos de aplicaciones y páginas web. Por lo tanto, esta aplicación me ha servido para realizar todo el prototipado de la aplicación.

4.3. Implementación detallada.

Ahora vamos a ver con más detalle todo lo que contiene nuestra aplicación.

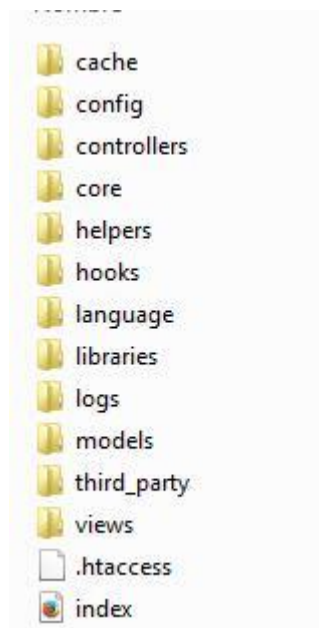


Ilustración 4.1. "Directorio archivos"



Config: en esta carpeta están todos los ficheros de configuración de la aplicación, entre ellos:

- **Autoload:** para cargar librerías, formularios, etc.
- **Config:** para poner URL como bases, cambiar configuración del proxy, etc.
- **Database:** fichero en el que se configura la base de datos.

- **Routes:** aquí se pone por defecto el controlador que queremos usar.
- ✓ **Controllers:** en esta carpeta se encuentran los ficheros con los distintos controladores de la aplicación, que como ya dijimos anteriormente hacen de unión entre las vistas y los modelos.
- ✓ **Models:** como su nombre indica, aquí se encuentran los ficheros de los distintos modelos que realizan las consultas a la base de datos.
- ✓ **Views:** aquí se encuentran todas las vistas de nuestra aplicación, lo que el usuario ve en pantalla realmente, y que muestra las consultas entre el controlador y el modelo.

En la raíz de la aplicación, también se encuentran archivos importantes para nuestra aplicación, las vemos a continuación:

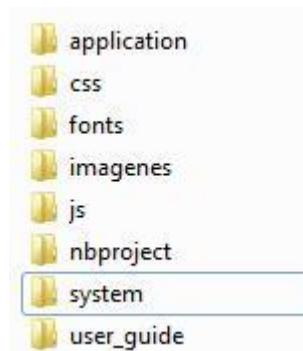


Ilustración 4.2. "Directorio CI"

- ✓ **Application:** aquí vemos los archivos y carpetas explicados anteriormente.
- ✓ **CSS:** en esta carpeta se encuentran todos los archivos que dan estilo y formato a nuestra aplicación web.
- ✓ **JS:** también incluyen archivos para el formato y estilo de la aplicación.
- ✓ **System:** en esta carpeta están archivos dedicados a la seguridad de la aplicación, librerías para otros tipos de lenguajes, otras opciones para la base de datos, etc.

Los archivos y carpetas no explicados no son importantes para la realización de la aplicación.

4.3.1. Configurar Servidor.

En mi caso el servidor usado para la realización de la aplicación ha sido WampServer, manteniendo el archivo de configuración del servidor tal y como estaba, lo único que se cambió para que funcionara correctamente fueron los puertos:

```
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80
```

Ilustración 4.3 “Configuración de puertos”

4.3.2. Conexión de la base de datos.

La base de datos usada en mi aplicación ha sido MySQL, y que es manejada por PHPMyAdmin.

La configuración de la base de datos se encuentra en el archivo config de la aplicación:

```

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'basedatos',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => TRUE,
    'db_debug' => TRUE,
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);

```

Ilustración 4.4. “Conexión base datos”

Como vemos aquí se configura la base de datos, pudiendo cambiar el nombre del usuario y contraseña de la base de datos (yo lo deje por defecto), el nombre de la base de datos que se haya utilizado en la aplicación, el tipo de codificación usado, en este caso, UTF-8, entre otras opciones.

Para trabajar con la base de datos, es decir, extraer datos de ella y almacenar datos, usamos en nuestro MVC los Modelos. Vamos a ver un pequeño ejemplo de cómo almacenar un alumno en la base de datos:

```

public function add_alumno()
{
    $usu = $this->session->userdata('username');

    $data = array(
        'nombre' => $this->input->post('nombre'),
        'apellidos' => $this->input->post('apellidos'),
        'curso' => $this->input->post('curso'),
        'user_name' => $usu
    );
    $this->db->insert('alumnos',$data);
    redirect(site_url().'/consultas/mostrar_alumnos');
}

```

Ilustración 4.5 “Modelo añadir alumno”

La función es simple lo que hace:

Primero almacena en la variable '\$usu' el usuario que está conectado en la aplicación actualmente.

Después en un array coge los datos del alumno introducidos por el usuario y los guarda en la variable '\$data'.

Por último introduce en la base de datos, en la tabla llamada alumno el array '\$data' con los distintos parámetros, y redirige a otra página.

4.3.3. Controladores de la aplicación.

Los controladores son los encargados de mandar a los modelos las consultas del usuario y después, mandar la respuesta del modelo a las vistas.

Para continuar con el mismo ejemplo vamos a ver la función del controlador que manda al modelo la consulta de añadir un nuevo alumno:

```
public function add_alumno()
{
    if($this->input->post('submit1'))
    {
        $this->form_validation->set_rules('nombre','Nombre','required');
        $this->form_validation->set_rules('apellidos','Apellidos','required');
        $this->form_validation->set_rules('curso','Curso','required');

        $this->form_validation->set_message('required','El campo %s es obligatorio');

        if($this->form_validation->run() != FALSE){
            $this->consultas_model->add_alumno();
            $data = array('mensaje' =>'El alumno se registro correctamente');
            $this->load->view('alumnos_view',$data);
        }else
        {
            $this->load->view('alumnos_add_view');
        }
    }
}
```

Ilustración 4.6. “Controlador añadir alumno”

Lo primero que hace la función es una condición para ver si se ha pulsado un botón de envío de información con el nombre 'submit1'. Si existe, realiza una

validación de los datos introducidos para ver si son correctos. Si no lo son, aparecerá un mensaje indicando cual es el error. Si ha comprobado que los datos son correctos, los manda a la función `add_alumno ()` del modelo vista anteriormente y recarga la página de los alumnos con un mensaje de éxito. Si no son correctos los datos, aparece de nuevo la página de añadir un nuevo alumno.

4.3.4. Vistas de la aplicación.

Esta parte es la que realmente ve el usuario, en la que aparecen los distintos formularios para introducir datos, las distintas barras de navegación para que el usuario seleccione la deseada, etc. Son las vistas a las que se le introduce el código CSS y JS para darle formato y estilo a las páginas y para que muestre distintas alertas cuando sea necesario.

Veamos un ejemplo:

```
<ul class="nav nav-tabs">
<li role="presentation"><a href="{% site_url 'welcome/' %}">Home</a></li>
<li role="presentation"><a href="{% site_url 'consultas' %}">Mi Perfil</a></li>
<li role="presentation"><a href="{% site_url 'consultas/mostrar_alumnos' %}">Alumnos</a></li>
<li role="presentation"><a href="{% site_url 'login/logout' %}">Desconectarse</a></li>
</ul>

<div class="container">
  <div class="row">
    <div class="col-md-4 col-md-offset-4">
      <h1>Insertar Alumno</h1>
      <form action="{% site_url 'consultas/add_alumno' %}" name="form" method="POST">
        <input type="text" name="nombre" id="alumno" placeholder="nombre"/><br/>
        <input type="text" name="apellidos" id="alumno" placeholder="apellidos"/><br/>
        <input type="number" name="curso" id="alumno" placeholder="curso"/><br/>
        <input type="text" name="user_name" id="alumno" placeholder="profesor"/><br/>
        <input type="submit" value="Enviar" name="submit1" />
      </form>
      <div class="alert-danger">{% validation_errors %}</div>
    </div>
  </div>
</div>
```

Ilustración 4.7. “Vista añadir alumno”

En la primera parte vemos la barra de navegación para volver atrás o para visitar cualquiera de las pestañas que vemos ahí: Home, Mi perfil, Alumnos (actual página) y Desconectarse del sistema.

En la segunda parte se ve el formulario para insertar un nuevo alumno. En el 'action' del formulario se pone la función del controlador a la que va a enviar los datos, en este caso a la función `add_alumno` del controlador `consultas`.

Después aparecen los distintos 'input' donde se van a introducir los datos requeridos.

Y por último aparece el 'submit' que es el botón encargado de enviar los datos al controlador.

Más abajo aparece una alerta en caso de que los datos no hayan sido validados por el controlador.

4.3.5. Estilo y formato CSS.

Ahora voy a explicar un poco más detalladamente con un ejemplo en qué consisten los CSS o también llamados estilos en cascada.

Los CSS no son más que líneas de código que dan formato a las distintas partes del código de HTML. Por ejemplo se le puede cambiar el estilo con CSS al 'button' por defecto de HTML.

Veamos un pequeño ejemplo:

```
<div class="contenedor" id="dos">
  <a href="<?=site_url('consultas/')?>"><img class="icon
  <p class="texto"> Mi perfil</p>
</div>
<div class="contenedor" id="cuatro">
  <a href="<?=site_url('consultas/mostrar_alumnos')?>">
  <p class="texto"> Alumnos</p>
</div>
```

Ilustración 4.8 "HTML estilo"

```
div#uno{  
    background-color: rgb(208,101,3);  
}  
div#dos{  
    background-color: rgb(233,147,26);  
}  
div#tres{  
    background-color: rgb(22,145,190);  
}  
div#cuatro{  
    background-color: rgb(22,145,100);  
}
```

Ilustración 4.9. "CSS estilo"

Como vemos en la primera figura el bloque de contenido es un "div" y se le asocia a uno el identificador dos y al otro el cuatro.

En la siguiente figura lo que hacemos es cambiar el aspecto original de la etiqueta "div" poniéndole a cada uno el fondo de color diferente, para dar un estilo más atractivo a nuestra página.

4.3.6. Mensajes y alertas de JavaScript.

Para hacer un uso más sencillo a los usuarios he añadido mensajes advirtiendo a los usuarios algún error introduciendo datos en algún formulario, pero también diciendo que los datos introducidos son correctos.

Veamos un ejemplo:

```
<div class="alert-danger"><?= validation_errors();?></div>
```

Ilustración 4.12. "Validación errores"

Esto mostraría por pantalla los errores que el controlador ha visto al realizar la validación de un formulario.

También puede ser un mensaje de que el formulario se haya rellenado con éxito:

```
<h1>Registrar Usuario</h1>
<?php if(isset($mensaje)):?>
<div class="alert-success"><?=$mensaje?></div>
```

Ilustración 4.13 “Registro éxito”

En este caso también es el controlador el que al validar los datos manda cargar la vista junto con un mensaje que aparecerá por la pantalla, en este caso de “Registrado con éxito”.

Con JavaScript se pueden mostrar mensajes para la confirmación de alguna acción, por ejemplo al eliminar un usuario, veamos:

```
<script lenguaje="JavaScript">
function enviar()
{
    if(confirm('¿Desea borrar?')){
        return true;
    }else{
        return false;
    }
}
</script>
```

Ilustración 4.14 “Función JavaScript”

Con la función “onclick=“return enviar()” que se encuentra dentro del formulario que queremos confirmar antes de enviar, lo que hace la aplicación es mostrar una pantalla una vez pulsado el botón de envío del formulario, en la que se muestra un mensaje de que si queremos borrar realmente los datos. Si pulsamos aceptar, se borran y nos redirige a una página establecida, y si pulsamos cancelar nos mantiene donde estamos sin ningún cambio.

4.3.7. Evaluación de las asignaturas.

Los instrumentos usados para el cálculo de las notas medias han sido: la prueba escrita o examen, una prueba oral y la libreta, contando cada una de ellas un 50,40 y un 10 por ciento respectivamente.

Para calcular el valor de la rúbrica de cada indicador lo he realizado según el valor del promedio de las notas del examen, libreta y prueba oral, quedando así:

```
<td id="uno">
<?php
@$rubrica = 0;
if($prom>=8){
    $rubrica=4;
}else if($prom <8 && $prom >=6){
    $rubrica=3;
}else if($prom <6 && $prom >=4){
    $rubrica=2;
}else if($prom <4){
    $rubrica=1;
}
echo "$rubrica"
```

Ilustración 4.15 "Cálculo de rúbrica"

Como observamos, si el valor del promedio calculado para un indicador es mayor o igual a ocho, el valor de rúbrica es el mayor, cuatro. Si se encuentra entre ocho y seis, incluido este, el valor sería tres. Si es menor que seis y mayor o igual a cuatro, el valor de la rúbrica sería dos, y por último siempre que el valor del promedio del indicador decaiga de cuatro, el valor de la rúbrica será uno.

Por último, para calcular el promedio de las competencias claves, se divide la suma de las medias de los indicadores entre el número de veces que los indicadores contienen una determinada competencia clave, es decir, si hay 4 indicadores con la competencia CMT, pues se dividiría la suma de las medias donde aparece dicho indicador entre el número de veces que aparece dicha competencia, en este caso 4.

5. Evaluación y pruebas de la aplicación.

Para la evaluación de la aplicación web vamos a realizarla mediante la técnica de la Caja Negra.

Estas pruebas permiten mediante unos test saber si la aplicación funciona correctamente viendo que se usan todos los requisitos funcionales de la aplicación.

En estas pruebas se permite encontrar:

1. Funciones incorrectas
2. Errores en la visualización de la interfaz
3. Errores en el acceso a la base de datos, tanto en el almacenamiento como en la extracción de datos.
4. Errores en el rendimiento.

Para realizar las pruebas necesitamos una serie de datos de entrada, unos test a esos datos, y ver la salida que ofrece el sistema. Veamos las distintas pruebas que hemos realizado a la aplicación:

5.1. Registro Usuario.

Nombre	Prueba	Resultado
Registro_uno	Algunos campos vacíos	El sistema muestra que campos están vacíos y que son necesarios rellenar.
Registro_dos	Poner un usuario existente.	El sistema advierte de que el usuario ya existe en la base de datos.
Registro_tres	Poner un correo existente	El sistema nos dice que ese correo ya existe en la base de datos.
Registro_cuatro	Poner todos los datos correctamente.	Aparece un mensaje de registrado con éxito y ya se puede acceder al sistema.

Tabla 5.1. "Prueba Registro"

5.2. Login del Usuario

Nombre	Prueba	Resultado
Login_uno	Dejar los campos vacíos o datos incorrectos	El sistema muestra un mensaje diciendo que el usuario y/o contraseña son incorrectos
Login_dos	Introducir bien los datos	El sistema nos lleva a la página principal

Tabla 5.2. "Login Prueba"

5.3. Cerrar sesión.

Nombre	Prueba	Resultado
Desconectar	Pulsamosobrela pestaña desconectar	La aplicación nos lleva a la página de inicio de sesión

Tabla 5.3. "Cerrar Sesión"

5.4. Editar Perfil.

Nombre	Prueba	Resultado
Editar_perfil	Pulsamos sobre la pestaña de Mi perfil	La aplicación nos lleva a la página de editar el perfil, donde podemos ver nuestros datos personales.

Tabla 5.4. "Editar Perfil"

5.5. Modificar datos.

Nombre	Prueba	Resultado
Modificar_datos	Pulsamos dentro de Mi perfil sobre el botón editar y modificamos los datos personales.	Pulsamos en guardar y nos llevará al inicio de sesión de nuevo.

Tabla 5.5. "Modificar Perfil"

5.6. Borrar cuenta de usuario

Nombre	Prueba	Resultado
Borrar_1	Pulsamos dentro de mi perfil, sobre el botón borrar. Y al aparecer el mensaje para confirmar, pulsamos en cancelar.	Nos aparecerá de nuevo la pantalla de mi perfil, quedando el usuario sin borrar.
Borrar_2	Pulsamos dentro de mi perfil, sobre el botón de borrar. Pulsamos en aceptar en el mensaje de confirmación.	Se nos eliminará el usuario por completo, redirigiéndonos a la página de iniciar sesión.

Tabla 5.6. "Borrar usuario"

5.7. Acceder a consulta de alumnos.

Nombre	Prueba	Resultado
Alumnos_Acceso	Pulsamos desde la página principal la pestaña de Alumnos.	El sistema nos llevará a la pantalla para consultar alumnos.

Tabla 5.7. "Acceder alumnos"

5.8. Consultar alumnos.

Nombre	Prueba	Resultado
Consulta_1	Introducir un curso que no posee alumnos para un profesor.	El sistema nos muestra la página vacía, sin la información de ningún alumno.
Consulta_2	Introducir un curso para los que un profesor tiene alumnos introducidos.	El sistema nos muestra los diferentes alumnos de dicho profesor.

Tabla 5.8. "Consulta alumnos"

5.9. Introducir Alumno

Nombre	Prueba	Resultado
Introducir_alum1	En la pestaña de introducir un nuevo alumno, dejamos algunos de los campos vacío.	El sistema nos advierte que los campos deben estar rellenos.
Introducir_alum2	En este caso rellenamos todos los campos.	El sistema nos manda a la página de consultar usuarios y podemos comprobar que ya está introducido.

Tabla 5.9. "Prueba introducir alumno"

5.10. Borrar alumno.

Nombre	Prueba	Resultado
Borrar_1	Una vez consultados los alumnos, pulsamos sobre borrar, y en el mensaje de confirmación cancelamos.	El alumno no se borra y sigue apareciendo los alumnos consultados.
Borrar_2	Una vez consultados los alumnos, pulsamos sobre borrar, y aceptamos el mensaje de confirmación.	El alumno se borra del sistema y de la base de datos, apareciendo de nuevo la página para consultar alumnos.

Tabla 5.10. "Prueba borrar alumno"

5.11. Seleccionar Asignatura.

Nombre	Prueba	Resultado
Selección_1	Una vez nos muestran los alumnos consultados, seleccionamos uno de ellos y nos aparece la página de asignaturas. Seleccionamos inglés.	Nos aparece la pantalla de evaluación de la asignatura de inglés.
Selección_2	Esta vez, seleccionamos la página de Lengua.	Nos aparece la pantalla de evaluación de la asignatura de Lengua.

Tabla 5.11. "Selección asignatura"

5.12. Evaluar Asignatura.

5.12.1. Calculo de medias.

Nombre	Prueba	Resultado
Media_indicador	Introducimos dentro de las casillas de las notas de un indicador cualquiera un valor para cada instrumento evaluable y pulsamos Intro.	En la casilla de la media de dicho indicador aparece la media de los 3 valores introducidos correctamente.
Media_ponderada	Dejamos los mismos valores introducidos anteriormente.	En las casillas de media ponderada perteneciente a cada instrumento, nos aparece el valor ponderado de los datos introducidos.

Tabla 5.12. "Calculo de medias"

5.12.2. Calculo de rubricas.

Nombre	Prueba	Resultado
Valores_Rubrica	Se introducen los valores dentro de las casillas pertenecientes a cada indicador, en este caso (7, 8, 6). Pulsamos Intro.	En la casilla de Rubrica aparece un valor del 1-4 asociado a la media de dicho indicador. En este caso la media es 7, por lo que el valor de rubrica toma el valor 3.
Valores_Media_Rubrica	Se introducen los mismos valores anteriores. Este indicador tiene asociado la competencia CL. Pulsamos Intro.	En la media ponderada de rubrica de CL aparece la misma nota anterior, el 7.

Tabla 5.13. "Cálculo de Rubricas"

6. Conclusiones.

Una vez terminada la realización de la aplicación para evaluar en Educación Primaria por materias y por competencias, se pueden sacar varias conclusiones.

Desde hace unos años se está intentando cambiar la forma de evaluar las materias de la Educación Primaria, lo que conlleva un cambio en la forma de enseñar el profesor y en la forma de aprender el alumno.

La evaluación es un proceso, por lo tanto no hay que confundirla con un resultado definitivo de esta, la calificación, ni con una de las fases del mismo, la evaluación final.

Ahora la evaluación tendrá un carácter continuo. Por un lado la misma evaluación formara parte del proceso de enseñanza-aprendizaje del profesor-alumno. También tendrá que ver el proceso de aprendizaje en cada tramo del mismo. También se referirá tanto al resultado de todo este proceso como a la forma en que el alumno trabaja en este proceso y como va adquiriendo los conocimientos.

El referente de todo este cambio deben ser las Competencias Básicas.

La evaluación continua del grado de adquisición de las Competencias Básicas derivará en una valoración de las mismas en cinco grados: poco, regular, adecuado, bueno, excelente.

También hay que tener en cuenta los criterios de evaluación, que especifican el grado de consecución de las capacidades expresadas en los objetivos y grado de adquisición de las competencias básicas. Estos criterios deben ser muy concretos y evaluables.

Para evaluar las Competencias Básicas bastará con realizar lo que siempre se han hecho con las materias, es decir, si para saber si se alcanzan los objetivos de un curso y materia hemos pasado por comprobar si se han alcanzado los criterios de evaluación con ellos relacionados, para la evaluación de la

adquisición de cualquier Competencia será necesario analizar el grado de desarrollo de los Criterios que están con cada una de ellas ligados.

Por lo tanto antes de evaluar, habrá que ligar cada criterio de evaluación con cada una de las Competencias que evalúa. Si entendemos que cada Criterio evalúa determinados objetivos y que estos están relacionados con las Competencias que desarrollan, tenemos la relación Criterios-Competencias.

Por último se necesitan de Instrumentos de evaluación para poder llevar a cabo la evaluación por competencias. Estos instrumentos deben ser variados y objetivos. También deben ser continuos, es decir, deben permitir la evaluación continua. Por tanto hemos de pensar en instrumentos aplicables durante todo el proceso de enseñanza-aprendizaje, y no solo al final.

Será el docente quien elija los Instrumentos que mejor se adapten a la dinámica del proceso de enseñanza-aprendizaje para que la aplicación de los mismos sea fácil y ágil.

6.1. Valoración personal.

El Trabajo Fin de Grado que escogí a mediados de Febrero me planteó algunas dudas al principio, ya que suponía una aplicación en principio bastante extensa y que debía llevar a cabo solo, programando una aplicación web completa, cosa que durante la carrera no había realizado, solo había llegado a hacer pequeños prototipos.

Esta aplicación me ha puesto algunas dificultades en el camino, ya que no conocía tan a fondo el código PHP y mucho menos JavaScript aunque haya sido para pequeñas cosas.

Gracias a manuales y tutoriales he podido aprender más sobre este tipo de código y también realizar toda la conexión con la base de datos.

Algo que si tenía aprendido de una de las asignaturas de la carrera y que me ha ayudado a decantarme por la realización de la aplicación web fue el framework Codeigniter, con el que he realizado toda la aplicación.

Gracias a la realización de este Trabajo Fin de Grado he podido adquirir conocimientos más específicos sobre el uso del patrón de arquitectura MVC, con el que he salido contento por cómo funciona y el que he comprendido a la perfección. También me ha ayudado al uso de los estilos en cascadas CSS para dar formato y estilo a toda la aplicación.

Por lo tanto el resultado final a nivel personal ha sido bastante bueno. Como he dicho anteriormente he aprendido bastantes conceptos. Eso sí, viendo bastantes tutoriales y manuales que me han ayudado mucho a la realización de esta aplicación.

Todo esto aprendido me puede ayudar en un futuro, tanto en el mundo laboral como incluso en mi tiempo libre, si decido realizar alguna página o aplicación web.

6.2. Posibles mejoras en un futuro.

Esta aplicación, aunque sea un prototipo, se podrá mejorar en un futuro con varias modificaciones o ampliación de requisitos:

Mejora en la seguridad de la aplicación para evitar ataques.

Posibilidad de realizar tutorías on-line con los padres de los alumnos.

Guardar en la base de datos toda y cada una de la evaluación de cada alumno.

Añadir un perfil y foto de cada alumno.

Permitir que los padres puedan ver las evaluaciones de los hijos una vez terminado cada trimestre a través de Internet.

7. BIBLIOGRAFÍA

Alvarez, M. A. (2 de Enero de 2014). *Qué es MVC*. Obtenido de Desarrolloweb.com: <http://www.desarrolloweb.com/articulos/que-es-mvc.html>

Forment, S. C. (16 de Julio de 2012). *MVC*. Obtenido de Mi blog técnico: <https://miblogtecnico.wordpress.com/tag/asp-net-mvc-2/>

MVC (s.f.), obtenido de la Wikipedia:
https://es.wikipedia.org/wiki/Modelo%20%80%93vista%20%80%93controlador#MVC_y_bases_de_datos

Erich Gamma, Richard Helm, Ralph Johnson y John M. Vlissides.(1994). *Design Patterns*.

CSS (s.f.): <http://www.w3schools.com/>

Tutorial Codeigniter: <http://www.tutorialspoint.com/codeigniter/>

ANEXO I.

MANUAL DE INSTALACIÓN

A continuación vamos a ver como se instala todo lo necesario para llevar a cabo el desarrollo de nuestra aplicación.

1.1. Instalación del servidor para la máquina Local. WampServer.

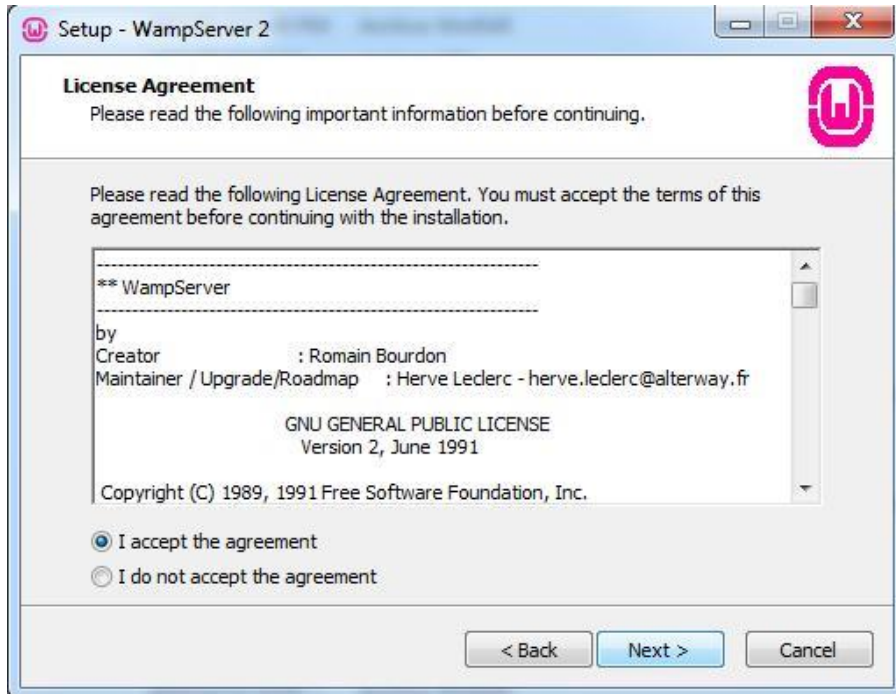
- a) Lo primero que haremos será descargar WampServer en nuestro ordenador. Para ello accedemos a la página oficial de WampServer y dentro de la sección de descargas, seleccionamos la opción según nuestro procesador.



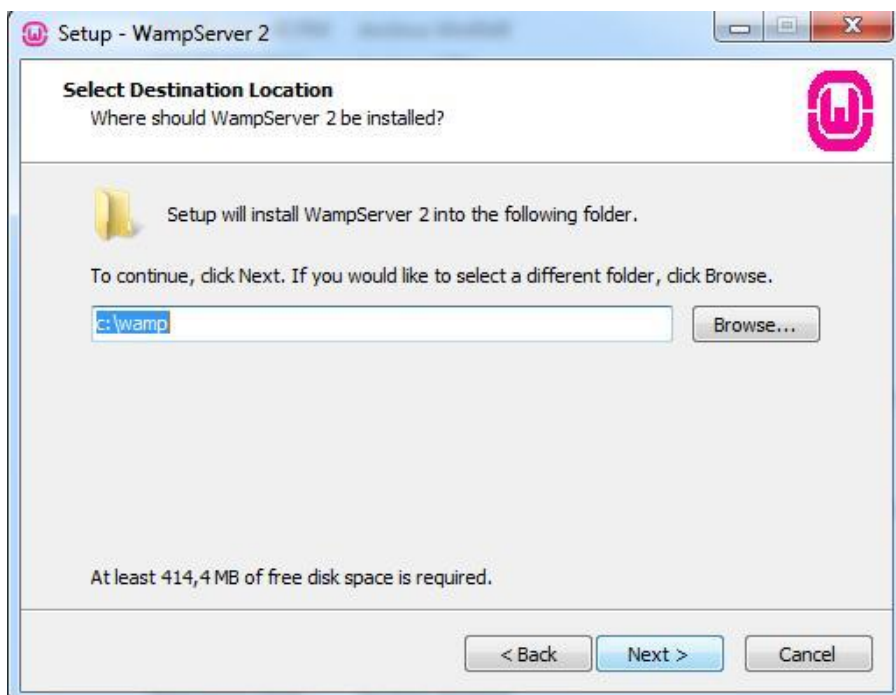
- b) Una vez lo tenemos descargado, lo ejecutamos y nos debe salir el Instalador, y pinchamos en siguiente o Next.



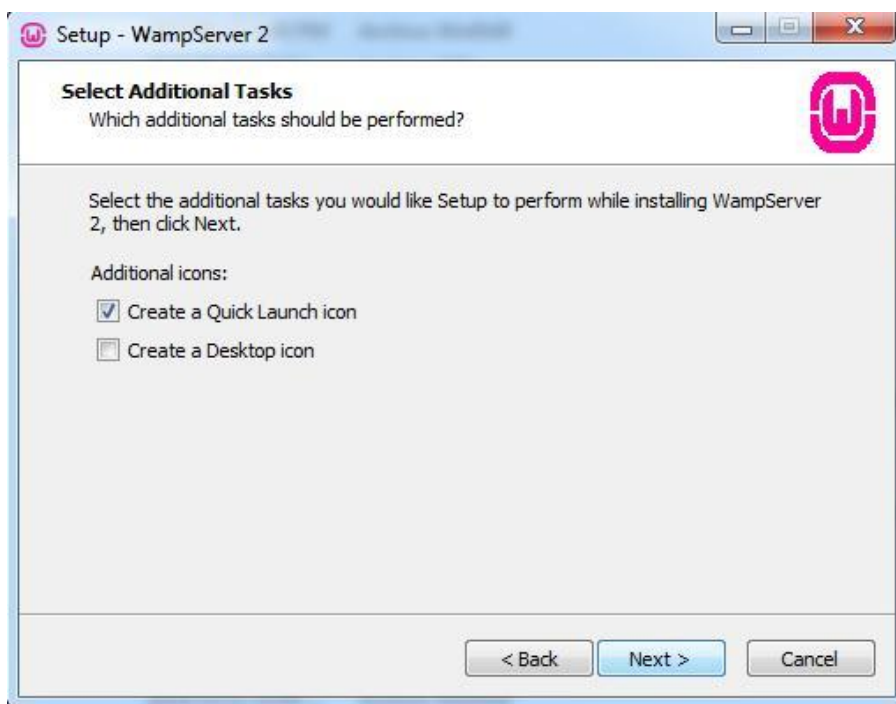
c) Aceptamos los términos y condiciones para seguir con la instalación.



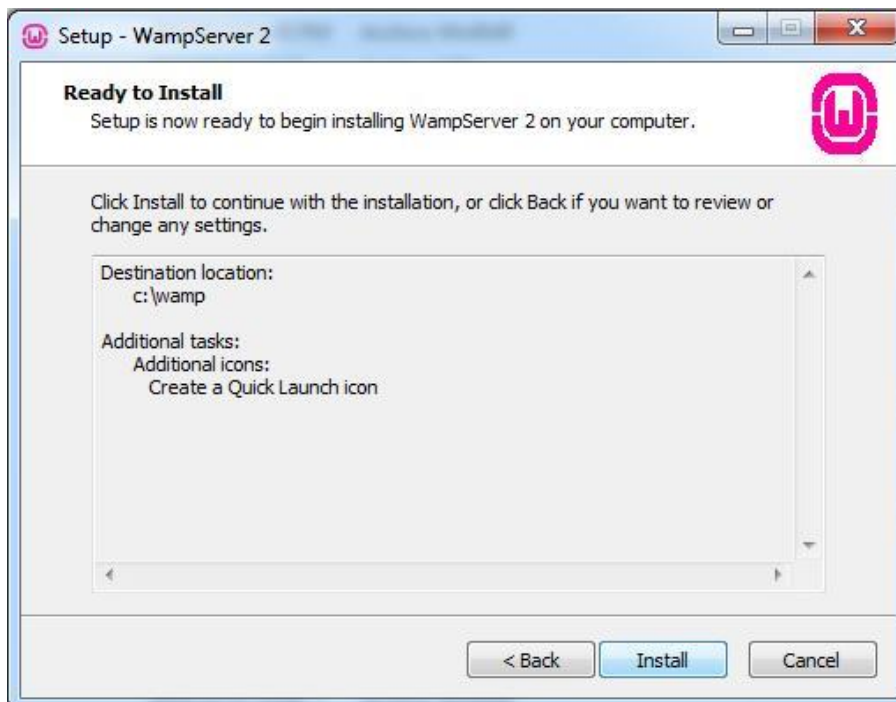
d) Seleccionamos la carpeta de destino de la instalación, en mi caso lo deje tal como venía. Pulsamos siguiente.



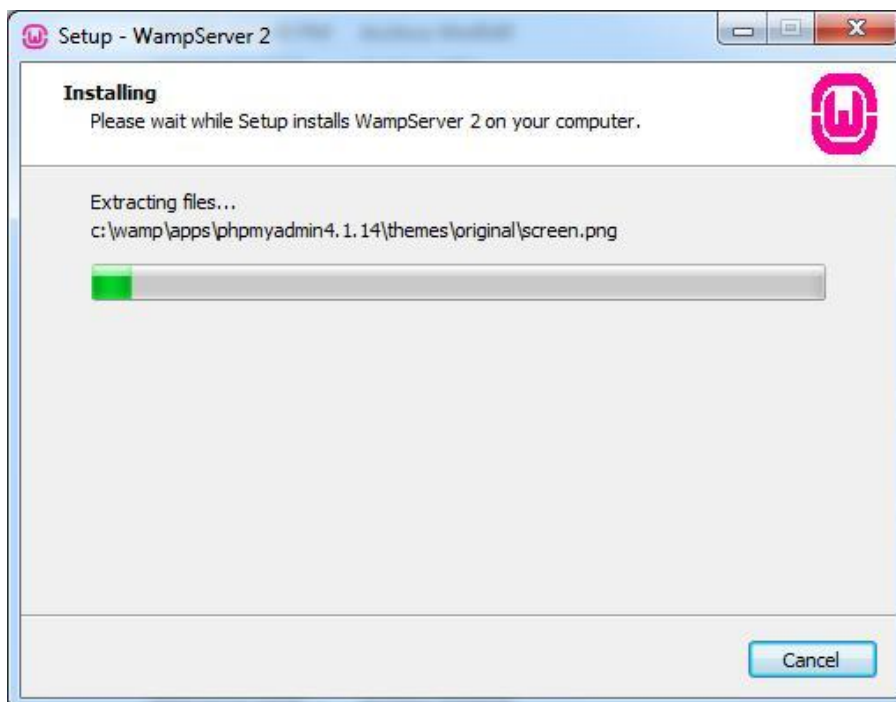
- e) Si se desea se puede poner un acceso directo. En mi caso deje la opción por defecto.



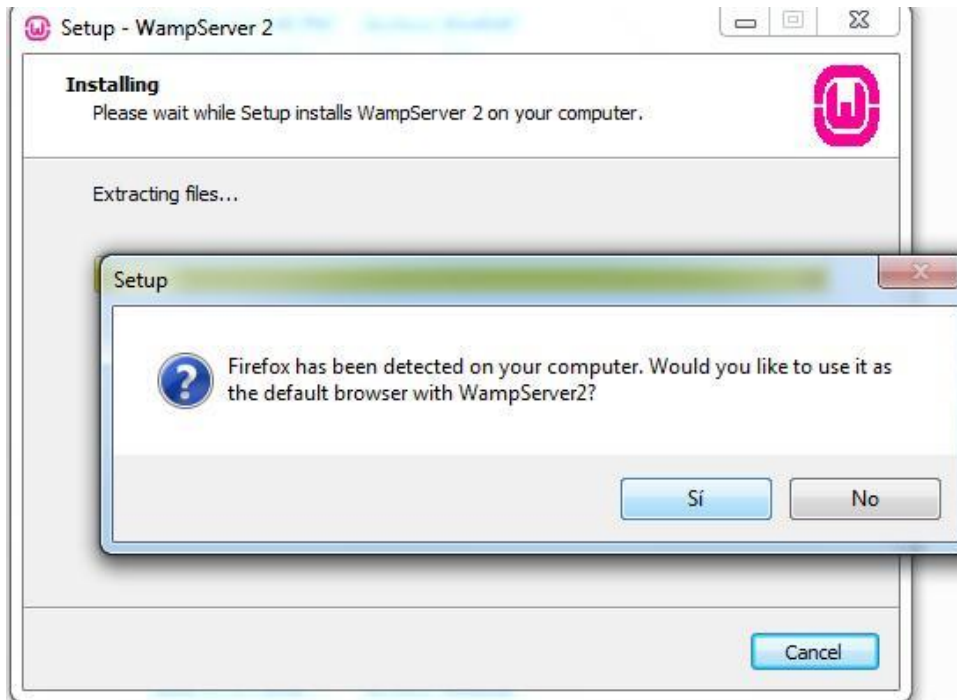
- f) Si todo va bien, pulsamos en siguiente, para continuar con la instalación.



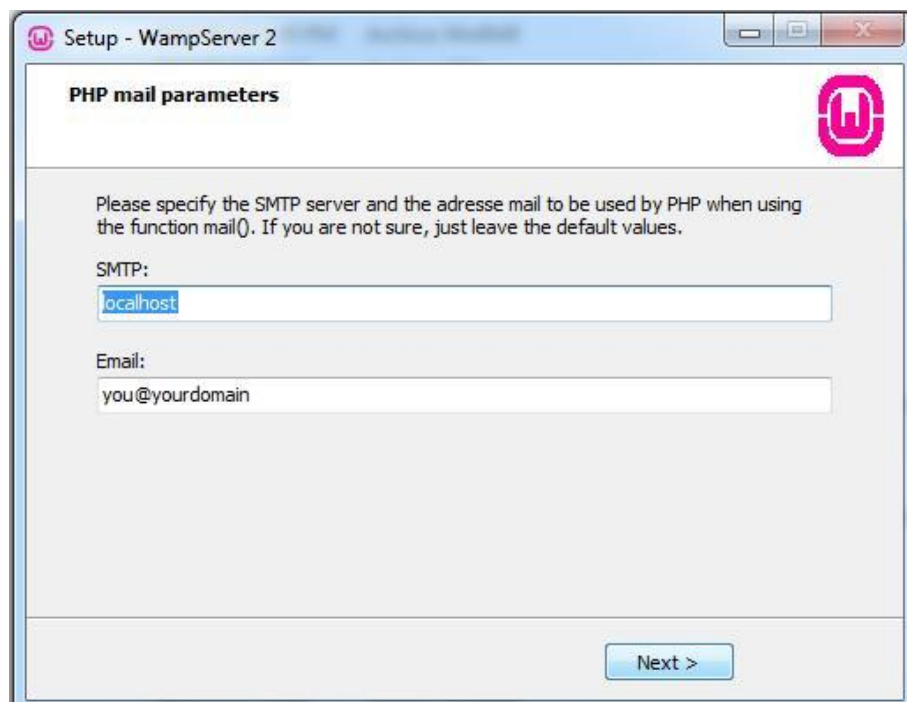
- g) La instalación comienza a realizarse.



- h) Después te detecta un navegador y te pide si lo quieres por defecto, en mi caso pulse No.



- i) Para terminar nos preguntará si queremos configurar el STMP. Yo lo dejo por defecto.



- j) El proceso de instalación ha concluido. Pulsamos en finalizar y se nos ejecutará el servidor en nuestra máquina local.



- k) El servidor se ejecuta en segundo plano en nuestra barra de tareas.



1.2. Instalación de Codeigniter.

1. Lo primero de todo debemos de descargar nuestro framework, en mi caso la versión más actualizada, directamente desde su página web, en la sección de descargas: <http://codeigniter.com/downloads/>
2. Después tenemos que subir los archivos al servidor, que en mi caso es la máquina local, copiándolos a la carpeta del servidor instalado anteriormente, llamada wamp/www.
3. Una vez tenemos la carpeta de la aplicación dentro de la carpeta del servidor local, le cambiamos el nombre que viene por defecto, en mi caso lo cambié por ci.
4. Una vez tenemos cambiado el nombre, debemos cambiar la URL base de nuestra aplicación para acceder a la raíz de Codeigniter.

Para ello abrimos el archivo de configuración que está en `system/application/config/config.php` con cualquier editor de texto y cambiamos la variable: `$config['base_url']`. Como Codeigniter lo he instalado en la máquina local, en esta variable he puesto: <http://localhost/ci> que es la carpeta cambiada de nombre explicada anteriormente.

5. Por último nos queda configurar la base de datos para poder consultar y almacenar datos. Dentro de la ruta `ci/application/config/` se encuentra el archivo `database.php` que tendremos que rellenar de acuerdo a los datos de nuestra base de datos.

Una vez tenemos instalado tanto el servidor en nuestra máquina local, hemos realizado las pertinentes configuraciones de la base de datos y tenemos Codeigniter en nuestro ordenador, ya podemos comenzar a desarrollar nuestra aplicación.

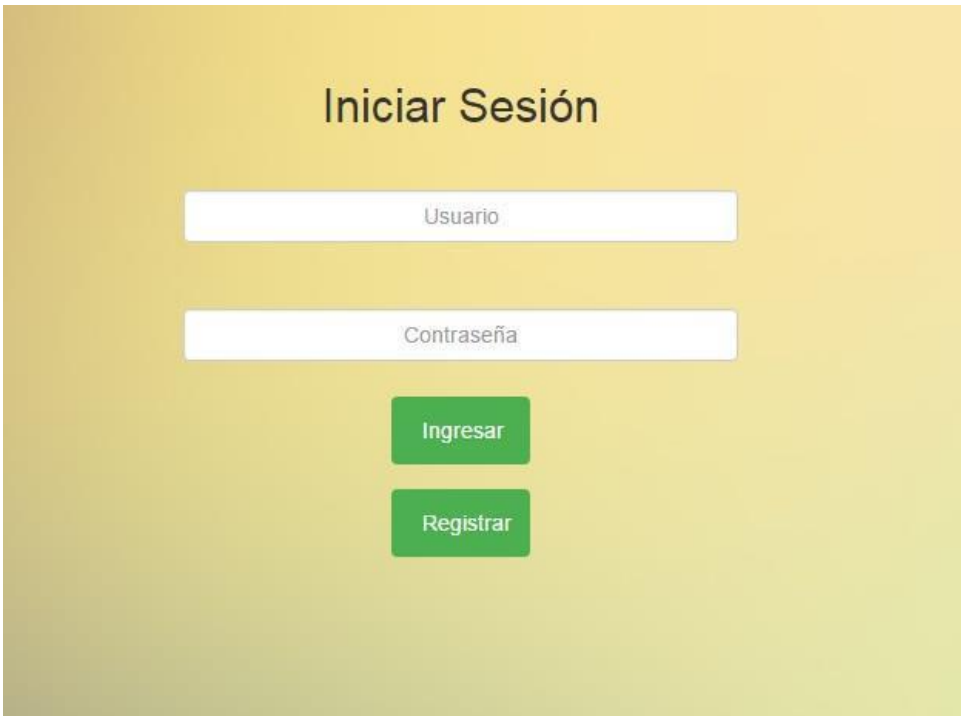
ANEXO II. MANUAL DE USUARIO

A continuación se va a exponer un manual de la aplicación con el objetivo de resolver cualquier duda que pueda tener el usuario durante su utilización.

2.1. Inicio de la aplicación.

Para poder iniciar la aplicación el usuario deberá introducir en el navegador: localhost/ci/index.php/login.

Una vez introducida aparecerá la pantalla para iniciar sesión en el sistema:

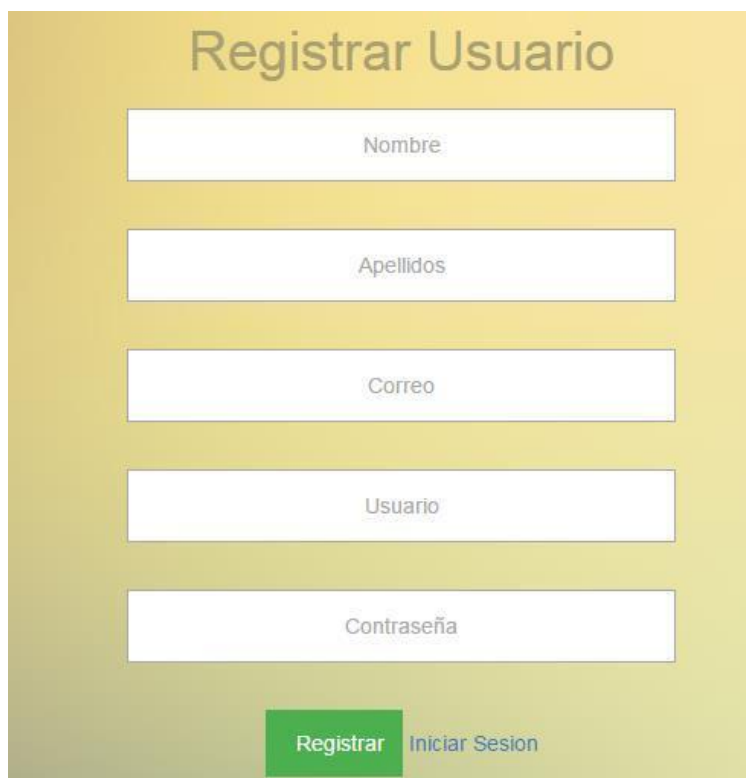


La imagen muestra una interfaz de usuario para iniciar sesión. El fondo es un degradado de amarillo a verde. En el centro, el título "Iniciar Sesión" está escrito en un color gris oscuro. Debajo del título, hay dos campos de entrada de texto blancos con bordes grises. El primer campo está etiquetado como "Usuario" y el segundo como "Contraseña". Debajo de los campos, hay dos botones rectangulares de color verde con texto blanco. El primer botón está etiquetado como "Ingresar" y el segundo como "Registrar".

Si el usuario se encuentra registrado, introducirá sus datos y le aparecerá la pantalla principal de la aplicación, si no lo está, deberá pulsar Registrar para crearse un nuevo usuario y poder acceder al sistema.

2.2. Registro Usuario.

Si el usuario no se encuentra registrado, pulsará Registrar y le aparecerá la siguiente pantalla donde rellenará el formulario.



Registar Usuario

Nombre

Apellidos

Correo

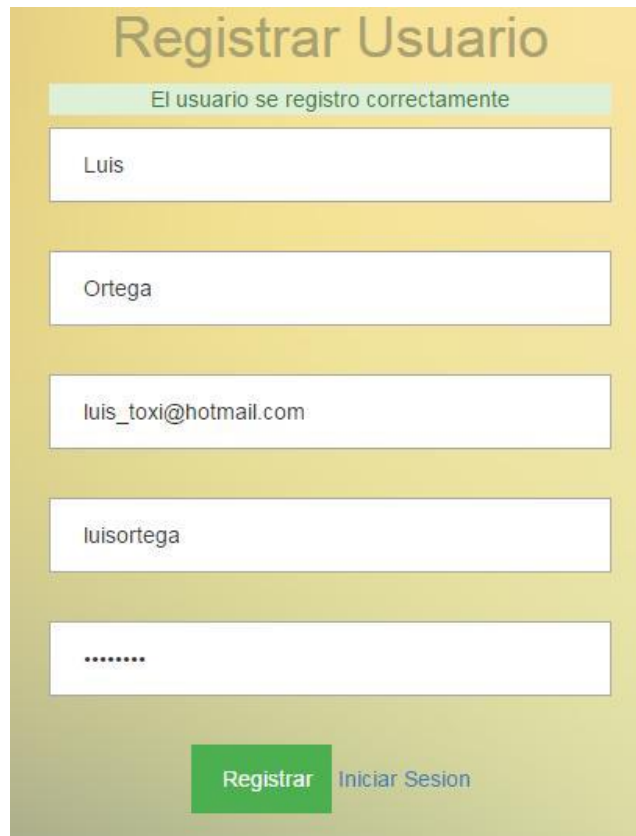
Usuario

Contraseña

Registrar Iniciar Sesión

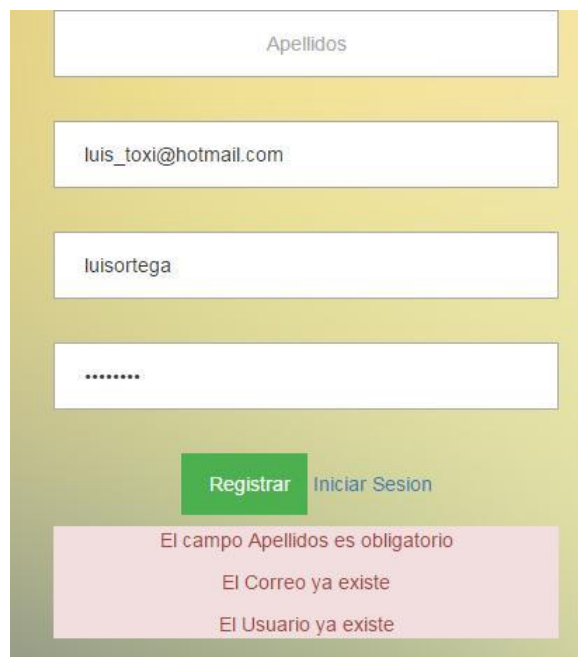
Detailed description: The image shows a registration form titled 'Registar Usuario' on a light yellow background. It contains five white input fields stacked vertically, labeled 'Nombre', 'Apellidos', 'Correo', 'Usuario', and 'Contraseña'. At the bottom, there are two buttons: a green 'Registrar' button and a blue 'Iniciar Sesión' button.

Si el usuario introduce los campos correctamente aparecerá un mensaje de confirmación:



The screenshot shows a registration form titled "Registrar Usuario" with a yellow background. At the top, a green message box says "El usuario se registro correctamente". Below this are five input fields: "Luis", "Ortega", "luis_toxi@hotmail.com", "luisortega", and a password field with seven dots. At the bottom, there are two buttons: a green "Registrar" button and a blue "Iniciar Sesion" button.

En cambio si alguno de los campos son introducidos erróneamente o existen en la base de datos aparecerá cuales tiene que cambiar el usuario.



The screenshot shows the same registration form as above, but with error messages. The "Apellidos" field is highlighted in red. Below the "Registrar" button, a pink message box contains the following text: "El campo Apellidos es obligatorio", "El Correo ya existe", and "El Usuario ya existe".

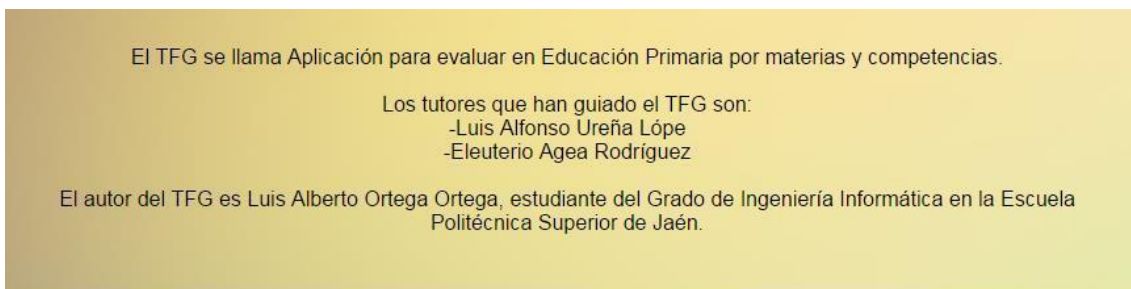
2.3. Pantalla Principal.

Una vez estamos registrados, ya podemos iniciar sesión. Si introducimos los datos erróneamente nos saldrá un mensaje de advertencia diciéndonos que el usuario y/o contraseña son incorrectos. Si los introducimos bien nos llevará a la página principal.



2.4. Información.

La primera pestaña que nos aparece es la de Home. Si pulsamos sobre ella nos aparecerá información básica sobre el TFG



2.5. Mi perfil.

La segunda pestaña, Mi perfil, nos ofrece los datos personales del usuario conectado al sistema y también la posibilidad de eliminar la cuenta del usuario o de editar algunos datos personales.



2.5.1. Si pulsamos sobre Editar nos aparecerá una pantalla donde poder cambiar los datos personales.



The screenshot shows a web interface titled "Editar perfil". It contains four input fields for editing user data:

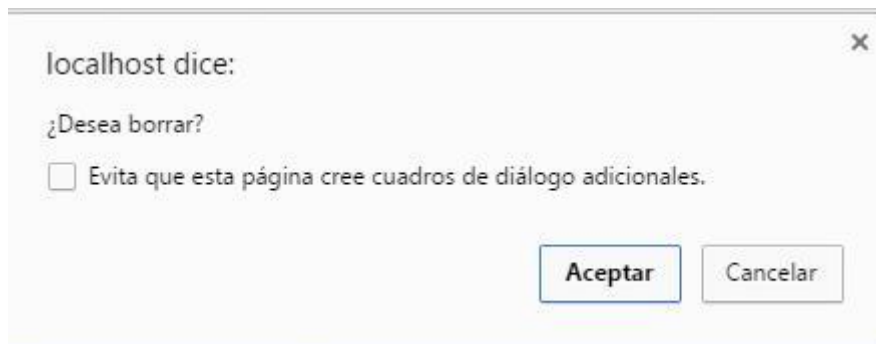
- Nombre: Luis
- Apellidos: Ortega Ortega
- Usuario: luisortega
- Contraseña:

Below the input fields is a green button labeled "Editar".

En este caso hemos cambiado añadiendo el segundo apellido. Pulsamos en editar y nos mandará a la página de iniciar sesión.

2.5.2. Borrar usuario.

Si pulsamos en la ventana de Mi perfil sobre Borrar, nos aparecerá un mensaje de confirmación por parte del sistema. Si pulsamos aceptar, se borrará y nos llevará a iniciar sesión y si pulsamos en cancelar, nos mantendrá en la pantalla actual.



2.6. Alumnos.

Si pulsamos dentro de la pantalla principal sobre alumnos, nos aparecerá lo siguiente:



Mis alumnos

Introduce curso

Consultar

Insertar

En esta pantalla tenemos dos opciones, Consultar alumnos por curso e insertar nuevos alumnos.

Veamos primero como se Insertan alumnos.

Pulsamos sobre Insertar y nos aparecerá la siguiente pantalla:



Insertar Alumno

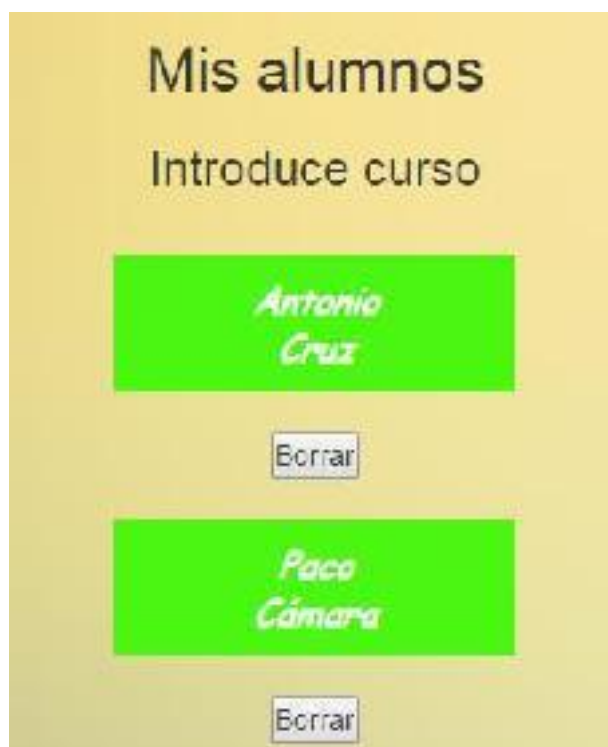
Enviar

Si nos dejamos algún campo vacío el sistema nos lo recuerda:



The image shows a web form titled "Insertar Alumno" (Add Student) on a yellow background. It contains three white input fields stacked vertically, labeled "nombre", "apellidos", and "curso". Below the fields is a green button labeled "Enviar". At the bottom, a light red box contains three lines of text: "El campo Nombre es obligatorio", "El campo Apellidos es obligatorio", and "El campo Curso es obligatorio".

Si se registra con éxito, dentro de la pantalla principal de alumnos, podemos consultar por curso y nos aparecerán los nombres de los alumnos.



Si pulsamos sobre el nombre, nos llevará a la página de las asignaturas para que sean evaluados (2.7).

En cambio sí pulsamos sobre el botón de borrar, nos aparecerá un mensaje de confirmación para ver si queremos eliminarlo o no del sistema.

2.7. Evaluación de los alumnos.

Como hemos dicho anteriormente, cuando pulsamos sobre un alumno, nos aparecerán las distintas materias en las que se puede evaluar el alumno en cuestión.



Cada una de estas asignaturas nos llevará a una tabla donde rellenar las notas, y poder evaluar al alumno tanto por materias como por competencias básicas.

Bloque 1. Matemáticas. 1º de Primaria. Antonio Cruz

Criterios de evaluación

C.E.1.1 Identificar y resolver situaciones problemáticas adecuadas a su nivel, partiendo del entorno inmediato, seleccionando las operaciones necesarias y utilizando razonamientos y estrategias. Aprender la utilidad de los conocimientos matemáticos que le serán válidos en la resolución de problemas. Expresar verbalmente de forma razonada y coherente el proceso seguido en la resolución, adoptando una respuesta coherente y abierta a debate.

C.E.1.2. Resolver situaciones problemáticas abiertas e investigaciones matemáticas sencillas sobre números, cálculos, medidas y geometría, iniciándose en el método de trabajo científico, utilizando diferentes estrategias, colaborando con los demás y explicando oralmente el proceso seguido en la resolución y las conclusiones. Utilizar medios tecnológicos para la búsqueda de información y realizar sencillos informes guiados para exponer el proceso y las conclusiones obtenidas.

C.E.1.3 Mostrar una disposición favorable hacia el trabajo matemático, valorando la presentación limpia y ordenada de los cálculos, así como confianza en las propias posibilidades y espíritu de superación de los retos y errores asociados al aprendizaje.

Indicadores	Examen	Libreta	Oral	Media	Rubrica	CCL	CMT	CD	CAA
MAT.1.1.1. Identifica, resuelve e inventa problemas aditivos de una operación en situaciones sencillas de cambio, combinación, igualación y comparación de la vida cotidiana. (CMCT).	<input type="text"/>	<input type="text"/>	<input type="text"/>	0.00	1		1		
MAT.1.1.2. Identifica los datos numéricos y elementos básicos de un problema, utilizando estrategias personales de resolución. (CMCT, CAA).	<input type="text"/>	<input type="text"/>	<input type="text"/>	0.00	1		1		1
MAT.1.1.3. Reconoce y asocia la operación que corresponde al problema. Expresa matemáticamente los cálculos a realizar y resuelve la operación que corresponde al problema, bien mentalmente, bien con el algoritmo de la operación o con calculadora. Comprueba la solución y explica con claridad el proceso seguido en la resolución. (CMCT, CAA).	<input type="text"/>	<input type="text"/>	<input type="text"/>	0.00	1		1		1

Rellenando los datos de esta tabla y pulsando sobre la tecla 'Intro'. Se rellenarán las casillas pudiendo ver las medias de cada indicador, la media total, la media ponderada y los valores de logro de cada una de las competencias.

ANEXO III. INDICE DE ILUSTRACIONES

Ilustración 1.1. Séneca	10
Ilustración 2.1. Planificador.....	12
Ilustración 2.2. Gantt	13
Ilustración 2.3. D. Registro	26
Ilustración 2.4. D. Inicio	26
Ilustración 2.5. D. Perfil	27
Ilustración 2.6. D. Borrar usuario	27
Ilustración 2.7. D. Insertar alumno.....	28
Ilustración 2.8. D. Consultar alumno	29
Ilustración 2.9. D. Borrar alumno	29
Ilustración 2.10. D. Asignatura.....	30
Ilustración 2.11. D. Evaluación.....	31
Ilustración 2.12. D. Desconectar	32
Ilustración 3.1. Arquitectura.....	
35 Ilustración 3.2. MVC	36
Ilustración 3.3. ER	38
Ilustración 3.4. Pr.Inicio	42
Ilustración 3.5. Pr.Registro	43
Ilustración 3.6. Pr.Principal.....	44
Ilustración 3.7. Pr.Perfil	45
Ilustración 3.8. Pr.Editar	45
Ilustración 3.9. Pr.Alumno	46
Ilustración 3.10. Pr.Consulta.....	47
Ilustración 3.11. Pr.Asignatura.....	48
Ilustración 3.12. Pr. Evaluar	49
Ilustración 4.1. Directorio Archivos.....	53
Ilustración 4.2. Directorio CI	54
Ilustración 4.3. Puertos	55
Ilustración 4.4. BBDD	56
Ilustración 4.5. Modelo	56

Ilustración 4.6. Controlador	57
Ilustración 4.7. Vista	58
Ilustración 4.8. HTML estilo	59
Ilustración 4.9. CSS estilo	60

