



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA INFORMÁTICA

Curso Académico 2009-10

Proyecto Final de Carrera

Aula Virtual: Gestor de Licencias

Autor

Guillermo Contreras Alcalde

Tutores

David Ríos Insua

Santiago Moral García

Resumen

El auge de las nuevas tecnologías de la información y de las comunicaciones está produciendo cambios significativos en todos los ámbitos de la sociedad. Por este motivo, la Universidad Rey Juan Carlos apuesta por una educación de calidad a través de su Campus Virtual, aprovechando las ventajas de las nuevas tecnologías para aplicarlas al campo de la educación a distancia, y adecuarse así a las demandas sociales.

Este Proyecto Final de Carrera es una aplicación web creada para utilizarse como herramienta del Campus Virtual de la Universidad Rey Juan Carlos. Esta aplicación servirá de apoyo técnico para la descarga de software específico, con o sin licencia, los pasos que los alumnos tienen que dar para descargarlo y ejecutarlo, además de informar sobre las normas y condiciones legales para hacer uso de los mismos. La aplicación “Aula Virtual: Gestor de Licencias” (AVGL) dará servicio, por un lado, a todos los alumnos online que tengan asignaturas en las que se necesite el uso de un determinado software y, por otro, servirá de apoyo a los técnicos que tendrán que supervisar y controlar todas las licencias que se asignen.

Con AVGL damos una solución a lo que hasta ahora era un problema. El proceso de asignación de licencias era un proceso manual. Era un trabajo tedioso y pesado tanto para los alumnos como para los técnicos, y, además, no se podía asegurar que el proceso se realizase de forma correcta.

Agradecimientos

Todo el mundo sabe que todo gran éxito tiene detrás un grupo de personas, que, aunque no salgan en la portada de este proyecto, sin su ayuda nunca se podría haber logrado. Todas esas personas que siempre estuvieron ahí, en los buenos momentos y en los no tan buenos, y que sin ellos estas líneas jamás se hubieran escrito. A todos ellos quiero dedicarles estas palabras.

En primer lugar quiero agradecerles a mi familia. Después de tantos años de gran esfuerzo, ellos siempre estuvieron a mi lado para animarme y creer en mí, porque sabían que aunque fuera difícil, yo podía conseguirlo. Papá, Mamá, gracias. Gracias por vuestro apoyo en todo momento, vuestra confianza, vuestra comprensión, vuestros consejos, en definitiva gracias por todo. Os quiero. A mi hermana Marta, sin duda un ejemplo a seguir. Sin ella seguramente nada de esto podría haber sido posible. Sé que llegarás lejos porque eres increíble, ya sabes yo fui el boceto y tú la obra maestra. Espero poder ayudarte tanto como tú me has ayudado a mí. Te quiero “piecito”. A mis abuelos Titi y Tata porque han sido como unos padres para mí. Y al resto de mi familia gracias a todos.

A lo largo de mi vida, he conocido a mucha más gente con la que he compartido muchas vivencias y nuevas experiencias. Algunos de ellos han acabado convirtiéndose en personas indispensables y sin su amistad no podría ser quien soy. Por eso también quiero darles las gracias a mis amigos.

Para empezar quiero agradecerles a mis compañeros y amigos de la universidad, con los que tan buenos momentos he vivido durante todos estos años. Entre ellos están: Dani Afro, Jorge, Iffy, Chisty, Jesús y Busti, toda la gente del “Spaguetti Aplazado”, de la AFI, los profes de la academia Ática y las chicas: Ana, Laura, Vita, Aroa y Elena. Gracias a todos.

A los amigos de siempre, los de toda la vida, los que a pesar de los años siempre están ahí. Me refiero a: Adrián, “Obrero”, “Bruja”, “Luispi”, “Vizu” y “Vitty”.

También quiero darles las gracias a los buenos amigos que hice en las diferentes épocas de mi vida, tanto en el colegio, fútbol, trabajo, etc. En especial a José, a la gente de informática y a la “superpandi”, que conocí durante mi etapa en el Media Markt.

Y por último gracias a mis compañeros del Campus Virtual: Natalia, Belén, Samu, Silvia, Alex y Javi. Y por supuesto a mis tutores David y Santy, que con su ayuda he conseguido realizar este proyecto.

En definitiva, a todas esas personas que me han apoyado y han confiado en mí todos estos años, gracias de todo corazón.

Índice general

1. Introducción.....	pág.7
1.1.Presentación del problema.....	pág.8
1.2.Objetivos.....	pág.9
1.3.Método de trabajo.....	pág.10
1.4.Estructura de la memoria.....	pág.11
2. Especificación de requisitos.....	pág.13
2.1.Introducción.....	pág.13
2.2.Descripción general.....	pág.13
2.3.Obtención de requisitos.....	pág.14
3. Estudios Previos.....	pág.17
3.1.Modelos de proceso para Ingeniería Software.....	pág.17
3.2.Tecnologías para Desarrollo de Aplicaciones Web.....	pág.23
3.3.Bases de Datos.....	pág.27
3.4.LDAP.....	pág.30
4. Análisis y diseño.....	pág.32
4.1.Especificación de los casos de uso.....	pág.32
4.2.Análisis de los casos de uso.....	pág.34
4.3.Diseño de la arquitectura del Sistema.....	pág.41
4.4.Diseño de Clases.....	pág.42
4.5.Diseño de la Base de Datos.....	pág.43
5. Implementación y pruebas.....	pág.45
5.1.Lenguaje de programación.....	pág.45
5.2.Detalles de implementación.....	pág.45
5.3.Implementación de la arquitectura.....	pág.51

5.4. Estructura de directorios.....	pág.52
5.5. Pruebas.....	pág.53
5.6. Manual de usuario.....	pág.55
6. Conclusiones.....	pág.77
6.1. Trabajos futuros y mejoras.....	pág.77
6.2. Estimación de esfuerzos.....	pág.78
Bibliografía.....	pág.79

CAPÍTULO 1

1. Introducción

Durante las últimas décadas, las mejoras en las tecnologías de la información y de las comunicaciones han hecho que nuestra sociedad haya evolucionado enormemente. Como resultado de esta evolución, hoy vivimos en los que muchos ha denominan como la “sociedad de la información”. Una sociedad donde las nuevas tecnologías juegan un papel fundamental y donde el poder de la información se convierte en una pieza clave para todas las actividades sociales, culturales y económicas. En esta nueva sociedad, una de las herramientas más importante, y sobre la cual giran gran parte de todas estas actividades, es Internet.

Internet se ha convertido en una fuente de información inagotable. El problema que surge hoy en día es que ya no nos vale con saber de dónde podemos sacar información, sino que es necesario saber manejar, distribuir e interpretar dicha información y tener la capacidad de, gracias a esos nuevos datos, poder generar conocimientos y tomar decisiones. En este contexto en el que Internet es tan importante, cobra especial interés crear herramientas para que la información que necesitemos sea fácilmente accesible y pueda ser gestionada según nuestros intereses. Fruto de ello surgen las aplicaciones web. Una aplicación web es una aplicación que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador.

La popularidad de las aplicaciones web se debe a que basta con tener solamente un ordenador, un navegador web y acceso a Internet para poder usarlas. Además, la facilidad de uso, la habilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software son otras de las ventajas que las hacen ser tan populares. También, las páginas web pueden contener elementos que permiten una comunicación activa entre el usuario y la información. Gracias a esto los usuarios pueden acceder a los datos de modo interactivo, ya que la página responderá a cada una de sus acciones.

El presente proyecto final de carrera pretende aprovechar todos estos avances tecnológicos. De hecho, todo el trabajo se desarrolló en una beca de apoyo a proyectos de investigación del servicio de tecnologías de la información. Más concretamente, se realizó dentro del Campus Virtual de la Universidad Rey Juan Carlos.

1.1. Presentación del problema

El Campus Virtual de la URJC pone a disposición de los alumnos online una serie de recursos que necesitarán a lo largo del curso académico. Dichos recursos pueden obtenerse a través de la página para alumnos del Campus Virtual: <http://apoyocv.escet.urjc.es/campusvirtual/recursosparaalumnos/index.php>. La descarga de software específico es uno de los recursos imprescindibles que necesitan los alumnos para desarrollar ciertas actividades durante el curso. Gran parte de este software es de pago y la Universidad dispone de un número de licencias limitadas. Por esta razón, se necesita tener un control de cuantas licencias se reparten y a qué alumnos se les asignan.

Actualmente, el mecanismo para asignar licencias es un proceso muy tedioso y poco intuitivo. Debido a este problema, surgió la necesidad de crear una aplicación web que permita asignar licencias a aquellos alumnos online que cursen alguna de las asignaturas en las que el uso de un software específico es imprescindible para llevar a cabo la docencia. Esta nueva aplicación se conoce como AVGL.

Hasta ahora, el alumno debía rellenar un formulario online con sus datos personales. Al finalizar, debía guardar dicho formulario en su ordenador en formato PDF. Este documento, siempre y cuando el alumno no lo borrara de su ordenador, servía como justificante de solicitud de licencia. Después, tenía que enviar este documento por correo electrónico al Campus Virtual. Una vez se recibía el fichero con los datos del alumno, se comprobaba manualmente que dicho alumno no hubiera solicitado una licencia antes y, en ese caso, se le asignaba una licencia nueva. Ésta se enviaba al alumno a través de correo electrónico. Todo este proceso podía tardar varios días. Además, no se tenía ninguna garantía de que se realizase correctamente, ya que, como hemos dicho, todo se tenía que hacer a través del correo electrónico.

En cuanto a la descarga del programa se refiere, el alumno podía descargar el software solicitara o no la licencia, pero hasta que no obtuviera ésta, no podía usarlo. Con la nueva aplicación, se quiere evitar esta práctica y se pretende que el alumno siga unos pasos determinados, entre los que se encuentra la solicitud de una nueva licencia, antes de descargar el programa.

En definitiva, con la aplicación AVGL lo que se pretende es crear una aplicación web que permita solucionar todos los problemas mencionados anteriormente y, como resultado, permita la descarga de programas y asignación de licencias para todos aquellos alumnos online que estén matriculados en determinadas asignaturas de manera fácil, cómoda y sencilla y, además, sea capaz de ayudar a los técnicos a gestionar y administrar todo el proceso.

1.2. Objetivos

Como vimos en la anterior sección, la descarga de software específico y la gestión y asignación de licencias eran un problema, ya que parte del proceso se hacía manualmente y era un verdadero quebradero de cabeza llevar un control de todo lo que se hacía. Por lo tanto, el objetivo principal del presente Proyecto Final de Carrera es solucionar este problema.

Para conseguir la realización del objetivo principal se presentan un conjunto de objetivos parciales:

- Análisis de los requisitos del sistema. Distinguir entre requisitos funcionales y no funcionales y priorizar los primeros.
- Realización de los estudios previos necesarios
 - Modelos de Proceso para Ingeniería Software
 - Tecnologías para Desarrollo de aplicaciones web
 - Bases de datos
- Diseño de una aplicación web capaz de cumplir con los requisitos y que, además, tenga la calidad necesaria para que el sistema funcione de manera eficiente.
- Implementación de la aplicación diseñada.
- Pruebas. A continuación se realizarán una serie de pruebas que comprobarán que nuestro sistema funciona según lo preestablecido.

1.3. Método de trabajo

El método de trabajo seguido en la realización de este proyecto viene ligado con cada uno de los objetivos parciales que se fijaron anteriormente. Paralelamente a la consecución de estos objetivos, se irá realizando la presente memoria. Basándonos en estos objetivos, podemos dividir este proyecto en cinco fases diferentes que enumeramos a continuación.

- **Fase I. Análisis de requisitos:** Se fijarán una serie de reuniones con personal del campus virtual, que establecerán los diferentes requisitos funcionales de la aplicación. De estas reuniones, obtendremos una lista de requisitos que se priorizarán para dar más importancia a los requisitos indispensables que debe tener nuestra aplicación.
- **Fase II. Estudios Previos:** Se realizarán una serie de estudios previos que determinarán las tecnologías que se van a utilizar para la realización del proyecto. Analizaremos las distintas posibilidades que tenemos a la hora de realizar una aplicación web y veremos cuál es la que mejor se adapta a nuestro problema.
- **Fase III. Diseño de la aplicación:** Una vez se definan los requisitos, se debe realizar un diseño acorde con los mismos, con el que se consiga obtener la funcionalidad requerida.
- **Fase IV. Implementación de la aplicación:** Una vez tengamos un diseño que se adapte a nuestras necesidades y solucione nuestro problema, nos dispondremos a implementar nuestra aplicación web.
- **Fase V. Pruebas:** La última de nuestras fases, pero no por ello menos importante, es la fase de pruebas. Cuando terminemos de implementar la aplicación, esta debe pasar un periodo de pruebas para determinar la calidad y poder solucionar posibles errores, antes de pasar a explotación. En esta fase, además, realizaremos los manuales de usuario que permitirá a alumnos y técnicos la posibilidad de consultar el funcionamiento de la aplicación.

Una vez terminemos todas las fases en las que hemos dividido nuestro trabajo la aplicación pasará a explotación, la cual no se incluye como fase, ya que escapa los límites de

este proyecto. En explotación, la aplicación debe funcionar correctamente y se hará un trabajo de mantenimiento y mejoras.

1.4. Estructura de la memoria

En este apartado haremos una breve descripción de los restantes capítulos en los que se divide el presente documento.

En el capítulo 2, comenzaremos la definición de nuestro sistema. En él se hará una detallada especificación de requisitos. Obtendremos todos los requisitos funcionales y no funcionales requeridos.

En el capítulo 3, antes de empezar a detallar en profundidad el proyecto, explicaremos los estudios previos que se realizaron para elegir la tecnología y el modelo de proceso usado en el proyecto. Explicaremos las posibles alternativas y, finalmente, argumentaremos la elección definitiva.

En el capítulo 4, se incluirán el análisis y diseño de la aplicación. Nuestro punto de partida será la especificación de los casos de uso. Entorno a los casos de uso definidos realizaremos un análisis de estos y obtendremos los diagramas que definirán el diseño de nuestro sistema. Tras esto, definiremos el diseño de la arquitectura del sistema. Explicaremos en qué consiste y la razón de su elección. A continuación detallaremos el diseño de las clases. Para este punto utilizaremos un diagrama de clases de nuestro sistema. Para terminar este capítulo mostraremos el diseño de la B.D.

El capítulo 5, está dedicado a la implementación y las pruebas realizadas. Se empezará hablando sobre el lenguaje de programación utilizado para desarrollar AVGL. Después, reservaremos un punto para explicar los detalles más importantes de la implementación. Explicaremos como se realizó y añadiremos capturas con aquellas partes de código que se consideren más importantes en el desarrollo. Además, enseñaremos como se implementó la arquitectura. También tendremos un apartado donde explicaremos la estructura de directorios de nuestra aplicación. Uno de los puntos más destacables dentro de este capítulo es el de las pruebas realizadas. Aquí se explicará qué se probó y qué resultados se obtuvieron. Para terminar tendremos un manual de usuario, donde explicaremos de manera fácil y sencilla que funcionalidades tiene AVGL y cómo deben usarse.

El capítulo 6, está dedicado a las conclusiones obtenidas tras la realización del presente proyecto. Empezaremos hablando sobre trabajos futuros y posibles mejoras. Y por último, haremos una estimación del esfuerzo realizado en este proyecto.

CAPÍTULO 2

2. Especificación de requisitos

2.1. Introducción

2.1.1. Propósito

La presente sección determina las directrices a seguir para el desarrollo de este proyecto y sirve de guía para validar dicho desarrollo y verificar su completitud. También resulta útil para determinar si futuros proyectos pueden encontrar utilidad en este producto. Al mismo tiempo, marca las pautas propuestas para realizar un futuro trabajo de mejora.

2.1.2. Alcance

Tal y como se explicó anteriormente nuestro sistema persigue mejorar la asignación de licencias y poder llevar un control de las mismas. Los requisitos detallados a continuación engloban las necesidades que pretende cubrir nuestra aplicación.

2.2. Descripción general

2.2.1. Funciones del sistema

Las principales funciones del sistema se podrían resumir en dos. Permitir que los alumnos consigan descargarse un software específico a través de la asignación de licencias. Qué un técnico controle y gestione la asignación de licencias permitiéndole consultar y editar la base de datos de la aplicación.

2.2.2. Características de los usuarios

En esta aplicación tendremos dos tipos de usuarios. Por un lado tenemos un perfil de usuario para alumnos online, por otro, tendremos un perfil de usuario al que denominaremos técnico.

- **Usuario alumno:** El perfil del usuario no requiere que se tenga ninguna experiencia o conocimiento previo sobre la aplicación. Este tipo de usuario

utilizará la aplicación para conseguir software específico y consultar las licencias que le han sido asignadas.

- **Usuario técnico:** Este perfil requiere cierto conocimiento sobre la aplicación. No es necesario que el usuario tenga un nivel muy avanzado, pero es recomendable que el usuario tenga algún conocimiento informático para poder acceder, consultar y editar la base de datos.

2.2.3. Restricciones

Para poder ser usuario alumno y tener acceso a la aplicación es requisito indispensable que el alumno sea online y que, además, esté matriculado en determinadas asignaturas. Si no cumple alguno de estos requisitos, no tendrá acceso a la aplicación.

2.3. Obtención de requisitos

A continuación se mostrarán en una tabla los requisitos, tantos funcionales como no funcionales, que se han tenido en cuenta para llevar a cabo la aplicación.

Tabla 1: Requisitos funcionales

REQUISITO	NOMBRE	DESCRIPCIÓN
RF1	Autenticarse	La aplicación constará de una interfaz gráfica que permitirá autenticarse de manera segura en la aplicación mediante usuario y contraseña.
RF2	Solicitar licencia	La aplicación constará de una interfaz gráfica que permitirá solicitar licencias a los alumnos online.
RF3	Descargar Software	La aplicación permitirá la descarga de software por parte del alumno, una vez haya conseguido una licencia.

RF4	Consultar Licencias	La aplicación permitirá consultar las licencias que ya le han sido asignadas al alumno.
RF5	Insertar Datos B.D.	La aplicación constará de una interfaz gráfica que permitirá al técnico de la aplicación insertar datos en la base de datos.
RF6	Modificar Datos B.D.	La aplicación constará de una interfaz gráfica que permitirá al técnico modificar datos en la base de datos.
RF7	Eliminar Datos B.D.	La aplicación constará de una interfaz gráfica que permitirá al técnico eliminar datos en la base de datos.
RF8	Consultar Datos B.D.	La aplicación constará de una interfaz gráfica que permitirá al técnico de la aplicación consultar datos en la base de datos.

Tabla 2: Requisitos no funcionales

REQUISITO	NOMBRE	DESCRIPCIÓN
RNF1	Usabilidad	La usabilidad de la aplicación debe ser adecuada en cuanto a los controles, iconos y ventanas.
RNF1.1	Tamaño de los botones, iconos, textos y ventanas	Los botones, iconos, textos y ventanas deben ser de un tamaño adecuado para su correcta visualización.

RNF2	Seguridad	La aplicación ofrecerá un servicio y acceso seguros.
RNF3	Escalabilidad	La aplicación debe ser construida con la idea de poder crecer en el futuro.
RNF4	Rapidez	La aplicación debe realizar un servicio rápido y eficaz.
RNF5	Compatibilidad	La aplicación debe ser compatible con diferentes navegadores y diferentes Sistemas Operativos.

3. Estudios Previos

En este capítulo se mostrarán los estudios previos que se han realizado para llevar a cabo este proyecto. Detallaremos algunas de las tecnologías más usadas en nuestros días y veremos el amplio abanico de posibilidades que tenemos para desarrollar nuestro proyecto.

3.1. Modelos de proceso para Ingeniería Software

Existe una multitud de modelos de proceso para Ingeniería de Software. Hablar de todos ellos llevaría un curso entero. Por eso, en este apartado realizaremos una breve descripción sobre algunas de las posibles alternativas que podríamos haber elegido y finalmente haremos especial hincapié en el modelo de proceso seleccionado.

➤ Modelo en Cascada

El modelo en cascada, también conocido como modelo clásico, ha sido uno de los más utilizados. Este modelo se caracteriza porque divide el proceso en etapas de forma que para poder iniciar una nueva etapa debe esperar a que se termine la etapa inmediatamente anterior.

Este modelo se puede dividir en cinco fases. Estas son:

1. **Definición de requisitos:** En esta fase se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada SRD (documento de especificación de requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos.

Es importante señalar que en esta etapa se debe consensuar todo lo que se requiere del sistema y será aquello lo que seguirá en las siguientes etapas.

2. **Diseño del Sistema y del software:** Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas

del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

3. **Implementación y prueba de unidades:** Es la fase de programación o implementación propiamente dicha. Aquí se implementa el código fuente, haciendo uso de prototipos así como pruebas y ensayos para corregir errores.
4. **Integración y prueba del sistema:** Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente y que cumple con los requisitos.
5. **Operación y mantenimiento:** El software obtenido se pone en producción. Se implantan los niveles software y hardware que componen el proyecto. La implantación es la fase con más duración y con más cambios en el ciclo de elaboración de un proyecto.

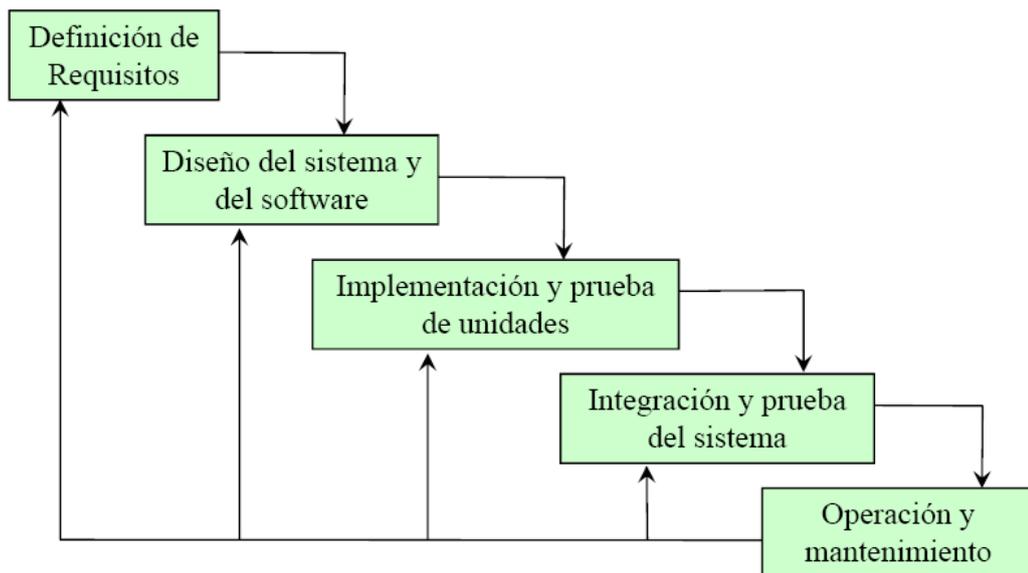


Figura 1: Modelo cascada

Ventajas

- Es un modelo apropiado sólo cuando los requisitos se comprenden muy bien.
- Se tiene todo bien organizado y no se mezclan las fases.

Inconvenientes

- Inflexibilidad al dividir el proyecto en etapas.
- Es difícil responder a los cambios en los requisitos del cliente.

➤ **Modelo en Espiral**

El modelo en espiral se propone en 1988 por Barry Boehm. El proceso se representa como una espiral más que como una secuencia de actividades con vuelta hacia atrás. En cada vuelta de la espiral se representa una fase del proceso. A diferencia del modelo en cascada, en el modelo en espiral no existen fases fijas. Cada vuelta en la espiral determina las actividades a realizar.

Cada ciclo del modelo en espiral estará dividido en cuatro actividades diferentes:

- 1. Definición de objetivos:** Se identifican los objetivos de cada fase, las alternativas y las restricciones.
- 2. Evaluación y reducción de riesgos:** Se determinan los riesgos de cada fase y se ponen en marcha las actividades que reduzcan estos riesgos.
- 3. Desarrollo y validación:** Se elige el modelo de desarrollo más apropiado para el sistema.
- 4. Planificación:** Se revisa el proyecto y, si se continúa, se planifica la siguiente fase (nueva vuelta a la espiral).

Ventajas

- Reduce los riesgos del proyecto.
- Incorpora objetivos de calidad.
- Integra el desarrollo con el mantenimiento.

Inconvenientes

- Genera mucho tiempo en el desarrollo del sistema
- Modelo costoso
- Requiere experiencia en la identificación de riesgos

En la siguiente figura se puede ver cómo se desarrolla un modelo en espiral.

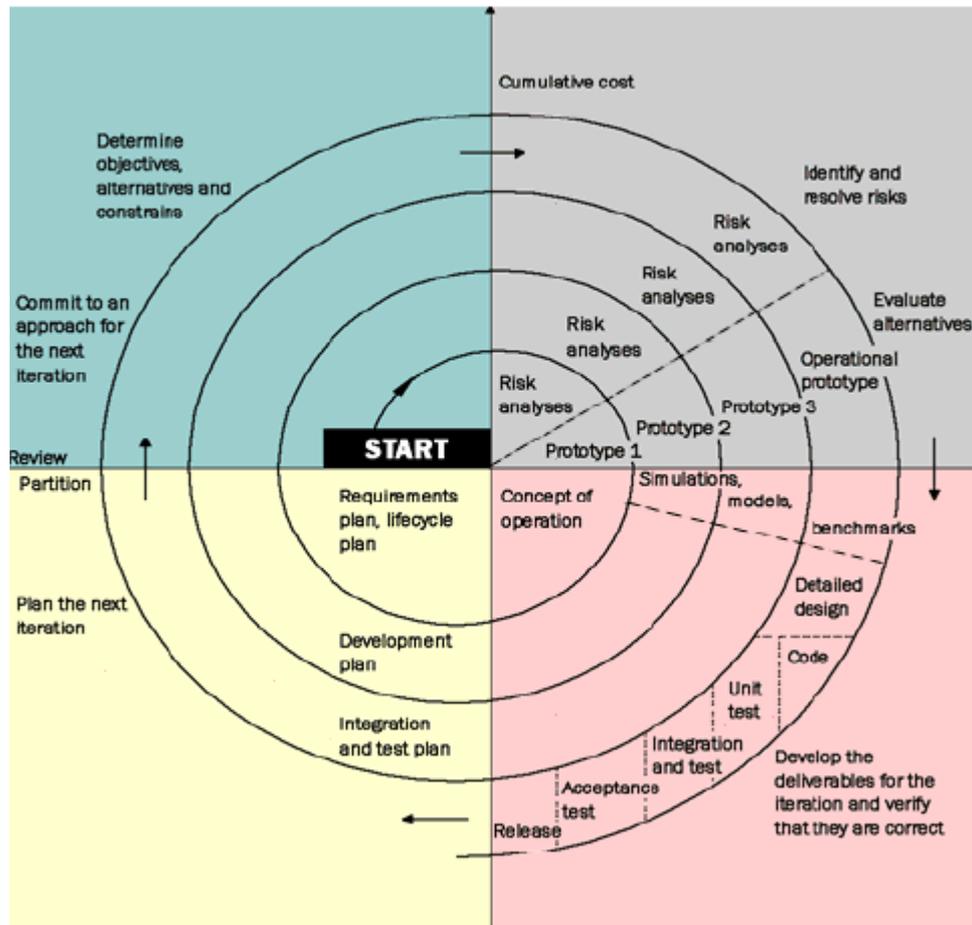


Figura 2: Modelo espiral

➤ Proceso Unificado

El último modelo de desarrollo del que vamos a hablar es el proceso unificado. Este es el modelo que hemos elegido para desarrollar nuestro trabajo.

El Proceso Unificado de Desarrollo Software (PU) es un marco que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental.

El PU puede verse como una metodología adaptable. Esto quiere decir que se puede modificar para adaptarlo al sistema concreto que se va a desarrollar en cada momento. Por otra parte se puede decir que el PU es una técnica para elaborar modelos que se adapta especialmente a UML. Su objetivo es producir un software de calidad. Este proceso no sólo considera aspectos de desarrollo de un sistema, sino también los de gestión del mismo.

El proceso unificado se caracteriza por los siguientes aspectos:

- **Iterativo e Incremental:** El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases está a su vez dividida en una serie de iteraciones. Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.

Cada una de estas iteraciones se divide a su vez en flujos de trabajo: Requisitos, Análisis, Diseño, Implementación y Pruebas. Aunque todas las iteraciones suelen incluir trabajo, el grado de esfuerzo dentro de cada flujo de trabajo varía a lo largo del proyecto.

- **Dirigido por los casos de uso:** En el PU los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de los distintos flujos de trabajo: requisitos, análisis, diseño, etc.
- **Centrado en la arquitectura:** El PU asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema. La analogía con la construcción es clara, cuando construyes un edificio existen diversos planos que incluyen los distintos servicios del mismo: electricidad, fontanería, etc.
- **Enfocado en los riesgos:** El PU requiere que el equipo del proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración, deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero.

Ventajas del desarrollo iterativo

- Hay varias oportunidades para revisar el sistema en estudio hasta que sea correcto.

- Se pueden encontrar errores y corregirlos.
- Adaptabilidad del desarrollo a nuevos requisitos o nuevos cambios.
- Se define una arquitectura sólida en etapas tempranas del desarrollo. La arquitectura de un sistema se define como un conjunto de componentes y las interacciones entre ellas. De este modo este tipo de ciclo de vida debe ser ampliable, por lo que el sistema es robusto, y tiene facilidad de mantenimiento.
- Se reducen los riesgos.
- En cada momento hay una versión del sistema funcionando que se modifica según las necesidades y deseos del cliente.

Inconvenientes

- Puede resultar un poco difícil para quien no tenga experiencia en el uso de procesos de ingeniería de software.

Por último en la figura que se muestra a continuación se pueden ver las distintas fases en las que se divide el PU y los flujos de trabajo que tiene cada fase.

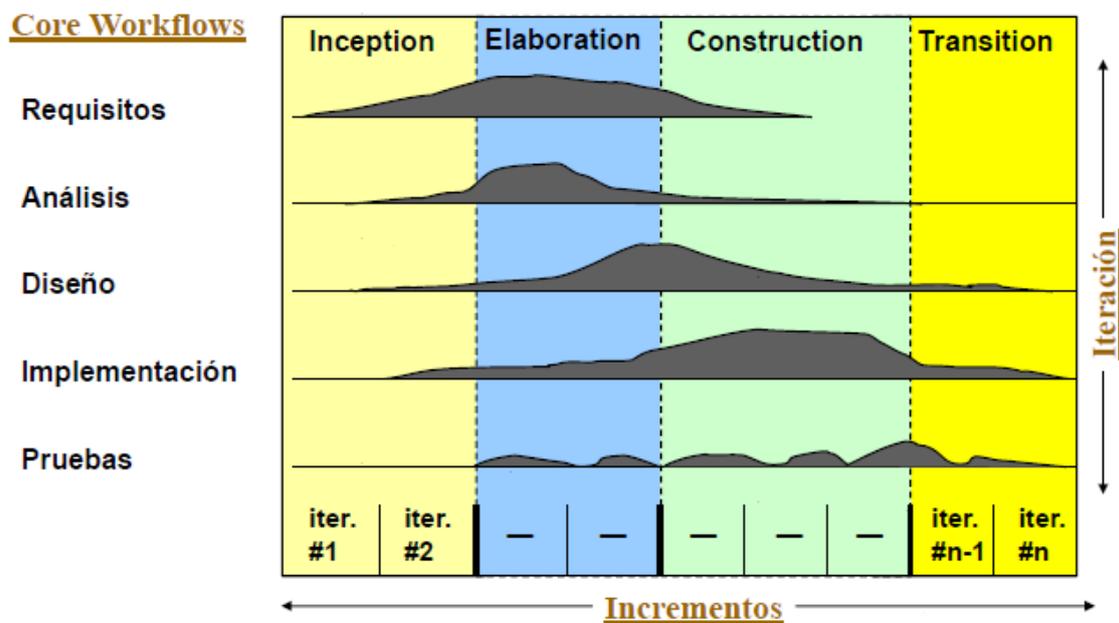


Figura 3: Proceso Unificado

3.2. Tecnologías para Desarrollo de Aplicaciones Web

Actualmente hay bastante variedad de tecnologías para el desarrollo de aplicaciones web. Es imprescindible en la fase de estudios previos hacer un análisis de las posibilidades que se presentan para la realización del proyecto. A continuación se resumen las tecnologías más extendidas y/o comerciales para la implementación web. Finalmente se prestará especial atención a las tecnologías que definitivamente se han decidido utilizar:

➤ Active Server Pages (ASP)

Es una tecnología impulsada y desarrollada por Microsoft. El entorno ASP consiste en páginas (archivos) que contienen texto, comandos y etiquetas HTML. ASP no es ningún lenguaje de programación en sí.

Ventajas

Desde el punto de vista de la empresa garantiza:

- Costes predecibles
- Fácil implantación
- Actualizaciones automáticas.

Además, permite el diseño de contenido web dinámico.

Inconvenientes

- Software propietario.
- Poco organizado.
- Se ejecuta sobre el servidor, lo que puede provocar mucha carga de procesos.

➤ ASP.NET

Es un conjunto de tecnologías para la implementación de páginas web impulsada y desarrollada por Microsoft. Destaca por la velocidad de respuesta del servidor y la seguridad. Esto último se consigue, en gran parte, por la distinción que hace la página entre código (donde, por ejemplo, se alojan las consultas a la base de datos) e interfaz de usuario.

Ventajas

- Velocidad
- Seguridad
- Contenido web dinámico
- Programación en capas.

Inconvenientes

- Software propietario.
- La licencia, las actualizaciones y el soporte son de pago.
- No es tan fácil encontrar documentación “libre” como las otras tecnologías GPL.

➤ Java Server Pages (JSP)

Es una tecnología Java orientada al diseño web dinámico y muy modular. Comparte algunas características con ASP.NET. Es posible ejecutar una aplicación que use tecnología JSP desde diferentes servidores sin necesidad de que tengan la misma plataforma software.

Ventajas

- Libre.
- Diseño web dinámico.
- Multiplataforma.
- Multiservidor.
- Orientado a objetos.
- Modular.
- Lenguaje de desarrollo Java.

Inconvenientes

- Hay motores libres pero la mayoría son de pago.
- Es más complejo y difícil de aprender.
- Se necesitan conocimientos del lenguaje y especificaciones Java.

➤ **Asynchronous JavaScript And XML (AJAX)**

AJAX no es una tecnología en sí misma, sino que combina tres tecnologías: HTML para el diseño, DOM (Document Object Model) como lenguaje script para interactuar con el interfaz que se le ofrece al usuario y XML como formato para la transferencia hacia el servidor.

Se define como una técnica de apoyo para crear aplicaciones web dinámicas e interactivas con el usuario. Mientras la aplicación se ejecuta en el navegador del usuario también se mantiene una comunicación asíncrona y en segundo plano con el servidor, así se pueden realizar cambios en la web sin necesidad de recargarla.

Ventajas

- Se ejecuta en la máquina cliente.
- Cada cambio no implica la recarga de la página completa.
- Mejor usabilidad para el cliente.
- Es fácilmente portable entre plataformas.

Inconvenientes

- Al cargarse dinámicamente la web provoca un comportamiento impredecible en el botón “volver” del navegador.
- Tampoco suele existir la posibilidad de guardar la URL ya que esta no identifica en que punto de la aplicación nos encontramos pues habrá sido cargada dinámicamente.

➤ **PHP Hypertext Pre-processor (PHP)**

PHP es una de las tecnologías elegidas para realizar este proyecto. Como hicimos con el resto de tecnologías haremos un breve resumen de las características del sistema y acabaremos explicando los motivos por los que hemos elegido esta tecnología.

Es un lenguaje de código fuente libre, sencillo de aprender, perfectamente integrado con HTML, utilizable en cualquier servidor (aunque nació pensado para Linux sobre Apache) y con una sintaxis familiar para desarrolladores C.

Ventajas

- Libre.
- Sencillo para el desarrollador.
- Comunicación con base de datos sin usar ODBC.
- Biblioteca muy completa.

Inconvenientes

- Lenguaje script, bajo nivel, no compilado.
- Limitado en la orientación a objetos.
- Se ejecuta sobre el servidor lo que puede provocar mucha carga de procesos.
- No genera contenido web dinámico por sí mismo, necesita apoyarse en otra/s tecnología/s.

En este proyecto se decide usar PHP sobre HTML principalmente por la facilidad de aprendizaje del lenguaje, por tener un API muy sencillo para la comunicación con las bases de datos SQL y Oracle con las que trabaja la aplicación y por ser una de las tecnologías más extendidas en su uso en el Área de Aplicaciones Corporativas y Microinformática.

➤ **JavaScript**

JavaScript es otra de las tecnologías que usaremos a lo largo del proyecto. A continuación presentamos algunas de sus características más importantes.

Es un lenguaje script (el más usado hoy) orientado a objetos. Está basado en acciones, poco restrictivo y generalmente incrustado como una pequeña funcionalidad dentro de una aplicación web. Es multiplataforma y se ejecuta por separado en las máquinas servidor y cliente.

No hay que confundir JavaScript con Java.

Ventajas

- Capaz de generar contenido dinámico.
- Se puede ejecutar en las plataformas más comunes usadas por los servidores.
- Seguro por el hecho de ser un lenguaje script.

Inconvenientes

- Disponibilidad de objetos limitada, lo que puede provocar la generación de scripts complejos para operaciones relativamente sencillas.
- El código es visible incluso el protegido con copyright.
- Debe cargarse por completo antes de ser ejecutado.

Principalmente se ha decidido usar JavaScript porque es muy útil para el control de errores en la propia aplicación.

3.3. Bases de Datos

Una base de datos (B.D.) nos permite guardar grandes cantidades de información de forma organizada y además nos posibilita encontrar y utilizar dicha información en el futuro. El término de B.D. fue escuchado por primera vez en 1963, en un simposio celebrado en California.

Desde el punto de vista informático, una B.D. es un sistema formado por un conjunto de datos que se encuentran almacenados en discos a los que se puede acceder directamente y un conjunto de programas que son capaces de manipular ese conjunto de datos. Cada B.D. se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más

columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla y cada fila de la tabla conforma un registro.

En el presente proyecto hemos usado dos tipos de sistemas de gestión de base de datos, que son Oracle versión 10g y MySQL versión 5.0.

➤ Oracle

Oracle es una herramienta cliente/servidor para la gestión de B.D. que tiene una gran potencia y se usa principalmente en grandes empresas.

Historia

Oracle surge a finales de los años 70 a partir de un estudio sobre SGBD (Sistemas Gestores de Base de Datos) de George Koch. El estudio se consideró como uno de los más completos sobre bases de datos. Usaba la filosofía de las bases de datos relacionales, toda una novedad en aquella época.

En la actualidad, Oracle todavía encabeza la lista y su tecnología se encuentra prácticamente en todas las industrias alrededor del mundo. Es la primera compañía de software que desarrolla e implementa software para empresas 100 por ciento activado por Internet a través de toda su línea de productos: base de datos, aplicaciones comerciales y herramientas de desarrollo de aplicaciones y soporte a la toma de decisiones.

Características de Oracle

Oracle es un sistema de gestión de base de datos relacional, desarrollado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Soporte multiplataforma.

Una B.D. Oracle tiene dos estructuras diferentes que son:

- **Estructura física:** se corresponde a los ficheros del sistema operativo. Una B.D. tiene uno o más ficheros de datos. Estos ficheros son de tamaño fijo y se establecen en el momento en que se crea la base de datos o en el momento en el que se crean tablespaces.

Los datos del fichero de datos son leídos cuando se necesitan y situados en una caché de memoria compartida para que el próximo acceso a los mismos sea más rápido.

- **Estructura lógica:** está formada por los tablespaces y los objetos de un esquema de B.D. Cada B.D. estará formada por uno o más tablespaces y cada tablespace se corresponde con uno o más ficheros de datos. Objetos: tablas, vistas, índices asociados a una tabla, clusters, etc.

➤ MySQL

MySQL es un SGBD relacional multiusuario, multiplataforma y con licencia GPL de la GNU. MySQL fue creado por la empresa sueca MySQL AB que conserva todo el código fuente del servidor bajo copyright.

Historia

SQL (Lenguaje de Consulta Estructurado) nació en IBM a principios de los 80's. Desde su publicación se adoptó como el estándar para las Bases de Datos Relacionales (BDR). En los 90's, aparecieron diferentes versiones del estándar y es al principio de la década cuando surge la necesidad de conectar tablas usando rutinas independientes del API de bajo nivel. La adaptación de estas rutinas se convirtió en la creación de funciones que, sin sacrificar fiabilidad, velocidad y usabilidad, terminaron por derivar en el interfaz del conocido SGBD denominado MySQL.

Cuando nació, MySQL carecía de elementos esenciales representables como transacciones y la regla de la integridad referencial. A pesar de este inconveniente consiguió levantar el interés de los desarrolladores atraídos por su simplicidad, por un diseño que le daba gran robustez y velocidad, y porque se pudieron parchear las carencias inherentes a las BDR por medio de las aplicaciones que la utilizan. Estas aplicaciones son numerosas pero las más conocidas que ofrecen conexión con MySQL en la actualidad son: C, C++, Pascal, Delphi, Java, Lisp, Perl, PHP, Python, Ruby, etc. Otra clave de su éxito, es que es soportado por gran cantidad de plataformas entre las más conocidas: Unix, Mac, Windows, Solaris, etc.

MySQL ha ido evolucionando a través de las sugerencias de los usuarios del sistema, canalizadas a través de la empresa creadora del proyecto, MySQL AB. Los desarrolladores de la empresa recopilan información de usuarios de Internet para ir ampliando la funcionalidad. Las versiones se van colgando en Internet a disposición de estos usuarios. No hay que olvidar distinguir las versiones beta (puestas a disposición de los usuarios para la explotación, detección de errores y para la propuesta de mejoras que posteriormente se aplican en la versión estable) de las estables que actualmente se encuentran en la v5.0.22

Características de MySQL

- Posibilidad de crear y configurar usuarios y sus permisos de acceso, lectura y escritura.
- Facilidad a la hora de exportar e importar datos o bases de datos completas.
- Posibilidad de ejecutar conjuntos de instrucciones (“query”) guardadas en ficheros externos a la base de datos.

Además de estas características generales a continuación se señalan las más importantes con las que cuenta la v5:

- Procedimientos almacenados, cursores, disparadores y vistas actualizables.
- Tipo de dato VARCHAR, INFORMATION_SCHEMA (tabla de configuración de la B.D. por defecto) y modo Strict.
- Soporte de transacciones distribuidas, motores de almacenamiento independientes y transacciones con dichos motores de almacenamiento.
- Cacheo de “query’s”, soporte SSL, SELECT’s anidados, indexación y soporte Unicode.

3.4. LDAP

Esta tecnología se utilizará en la autenticación de los usuarios, cuando quieran acceder a la aplicación. LDAP es una B.D. que se utiliza para guardar el usuario y la contraseña de los alumnos. El motivo de utilizar LDAP es que los usuarios pueden utilizar el usuario y contraseña de dominio único. Es decir, podrán usar el mismo usuario y contraseña que utilizan para acceder al resto de aplicaciones en las que tienen acceso dentro de la universidad.

LDAP (*Lightweight Directory Access Protocol*) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos a la que pueden realizarse consultas.

Habitualmente, almacena la información de autenticación (usuario y contraseña) y es utilizado para autenticarse aunque es posible almacenar otra información. A manera de síntesis, LDAP es un protocolo de acceso unificado a un conjunto de información sobre una red.

La versión actual es LDAPv3, la cual es especificada en una serie de Internet Engineering Task Force (IETF) Standard Track Request for Comments (RFCs) como se detalla en el documento RFC 4510.

Ventajas

- Es muy rápido en la lectura de registros
- Permite replicar el servidor de forma muy sencilla y económica
- Muchas aplicaciones de todo tipo tienen interfaces de conexión a LDAP y se pueden integrar fácilmente
- Dispone de un modelo de nombres globales que asegura que todas las entradas son únicas
- Usa un sistema jerárquico de almacenamiento de información.
- Permite múltiples directorios independientes
- Funciona sobre TCP/IP y SSL
- La mayoría de aplicaciones disponen de soporte para LDAP
- La mayoría de servidores LDAP son fáciles de instalar, mantener y optimizar.

Inconvenientes

- LDAP resulta complicado de configurar ya que es un sistema complejo.

CAPÍTULO 4

4. Análisis y diseño

4.1. Especificación de los casos de uso

Para comenzar este apartado, incluiremos un diagrama con los casos de uso y los usuarios que tienen acceso a la aplicación. En el diagrama se puede apreciar qué casos de uso irán asociados a cada uno de los usuarios y cuáles son comunes a los dos.

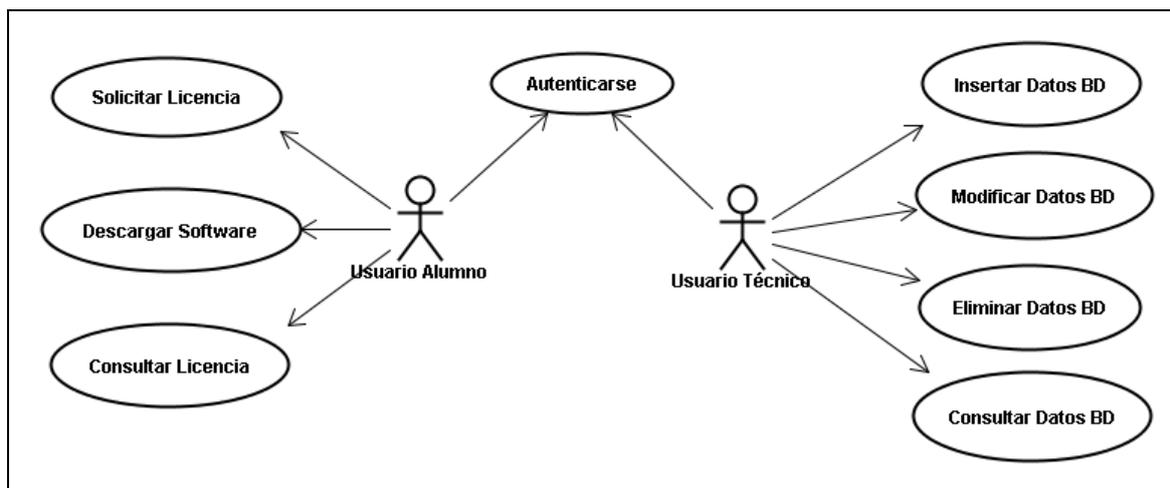


Figura 4: Diagrama de Casos de Uso

En función de la especificación de requisitos obtendremos los caminos estándar y alternativos. En las siguientes tablas, mostraremos sólo los caminos estándar para cada requisito, ya que los alternativos son variaciones de éstos. De esta forma, evaluamos el comportamiento de nuestra aplicación web en los diferentes casos de uso.

RF1: Autenticarse	
Usuario	Sistema
1. Solicita autenticarse	2. Solicita Datos
3. Introduce login y password	4. Valida Datos con B.D.

RF2: Solicitar Licencia	
Usuario	Sistema
1. Solicita Licencia	2. Solicita aceptación de los términos de la Licencia
3. Acepta términos de la Licencia	4. Presenta datos del alumno extraídos de la B.D.
5. Acepta datos del alumno	6. Confirma asignación de licencia
	7. Inserta la licencia asignada en la B.D.

RF3: Descargar Software	
Usuario	Sistema
1. Selecciona descargar Software	2. Solicita número de Licencia
3. Introduce número de licencia	4. Valida número de licencia con la B.D.
5. Inicia la descarga del Software	

RF4: Consultar Licencias	
Usuario	Sistema
1. Selecciona consultar licencias	2. Muestra licencias del alumno
3. Elige la licencia	4. Muestra la licencia seleccionada

RF5: Insertar Datos en B.D.	
Usuario	Sistema
1. Solicita insertar datos en B.D.	2. Solicita Datos para insertar
3. Inserta datos	4. Valida Datos
	5. Pide Confirmación para la inserción
6. Confirma inserción	7. Inserta datos en la B.D.

RF6: Modificar Datos en B.D.	
Usuario	Sistema
1. Solicita modificar datos en B.D.	2. Solicita Datos para modificar
3. Modifica datos	4. Valida Datos
	5. Pide Confirmación para la modificación
6. Confirma modificación	7. Modifica datos en la B.D.

RF7: Eliminar Datos en B.D.	
Usuario	Sistema
1. Solicita eliminar datos en B.D.	2. Solicita Datos para eliminar
3. Inserta datos de la eliminación	4. Valida Datos
	5. Pide Confirmación para la eliminación
6. Confirma eliminación	7. Elimina datos en la B.D.

RF8: Consultar Datos en B.D.	
Usuario	Sistema
1. Solicita consultar datos en B.D.	2. Solicita Datos para consultar
3. Inserta datos de la consulta	4. Valida Datos
	5. Muestra la consulta solicitada

4.2. Análisis de los casos de uso

4.2.1. Diagramas de análisis

Una vez realizado el estudio de los Casos de Uso identificaremos las clases que van a formar parte del sistema. Este primer análisis lo haremos a partir de diagramas UML de clases de análisis. Tendremos un diagrama por cada caso de uso.

Los diagramas tienen tres componentes básicos. Para empezar, tendremos una interfaz de usuario. A continuación tendremos un componente de control que será quien se

encargue de realizar la funcionalidad específica. Y por último tendremos la B.D. que será de donde obtendremos toda la información que se necesite para realizar una función concreta.

A continuación iremos mostrando cada uno de los diferentes diagramas de análisis que obtendremos de cada una de nuestras funcionalidades.

- **Autenticar**

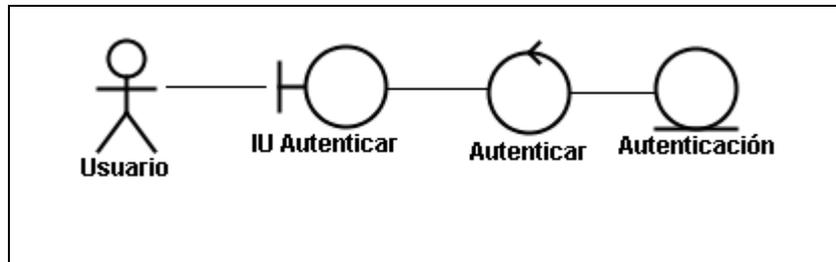


Figura 5: Diagrama de Análisis. Autenticar

- **Solicitar Licencia**

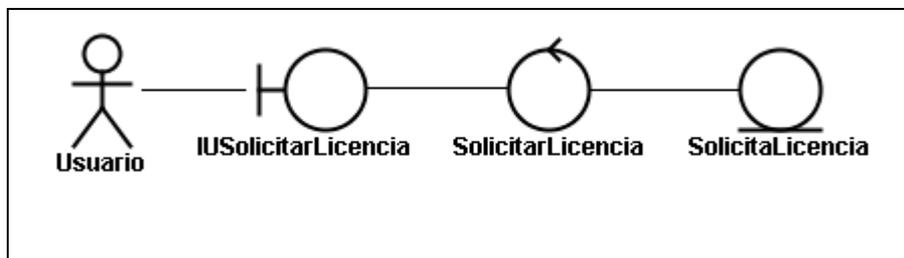


Figura 6: Diagrama de Análisis. Solicitar Licencia

- **Descargar Software**

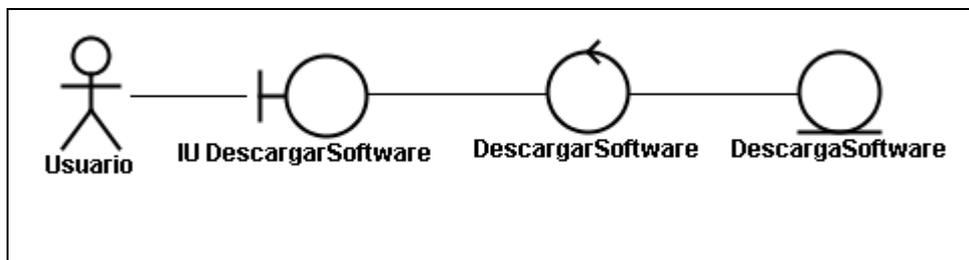


Figura 7: Diagrama de Análisis. Descargar Software

- Consultar Licencias

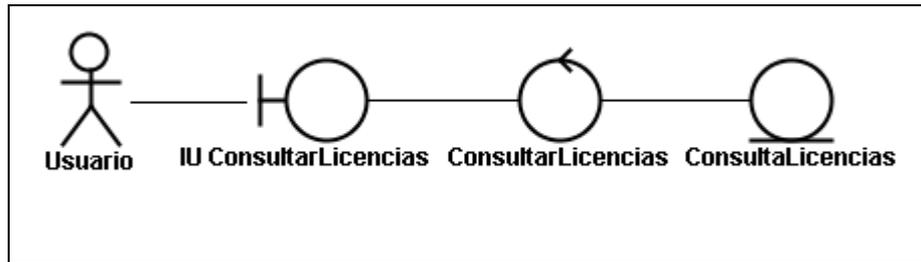


Figura 8: Diagrama de Análisis. Consultar Licencias

- Insertar Datos

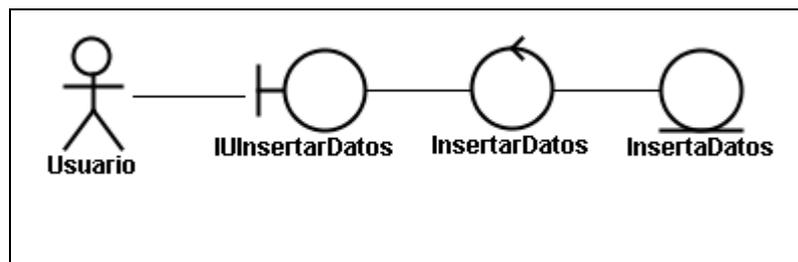


Figura 9: Diagrama de Análisis. Insertar Datos

- Modificar Datos

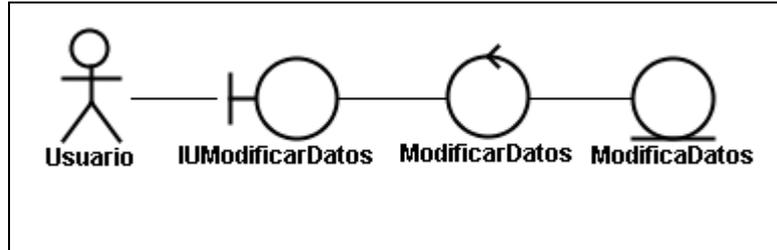


Figura 10: Diagrama de Análisis. Modificar Datos

- Eliminar Datos

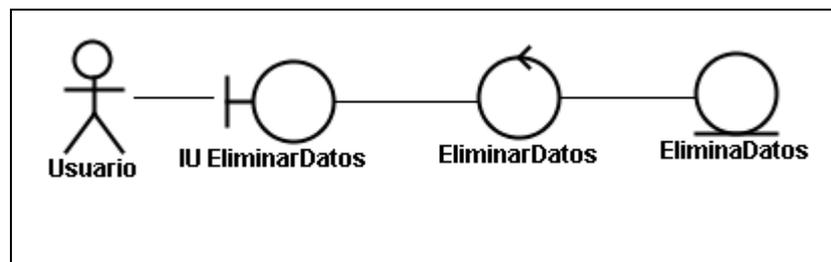


Figura 11: Diagrama de Análisis. Eliminar Datos

- **Consultar Datos**

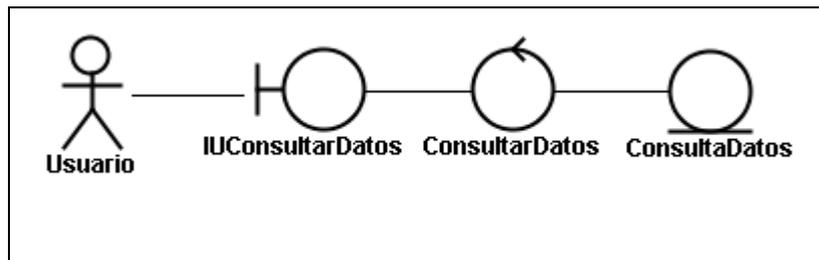


Figura 12: Diagrama de Análisis. Consultar Datos

4.2.2. Descripción de la interacción de objetos

A continuación pasaremos a describir la interacción de los objetos que hemos identificado en nuestro sistema. Para ello, utilizaremos diagramas de colaboración que describen el comportamiento del sistema a lo largo del tiempo.

➤ Diagramas de colaboración

En este punto, incluiremos los diagramas de colaboración que realizan las funcionalidades definidas en los casos de uso. Una colaboración es una interacción entre un conjunto de objetos para implementar un comportamiento del sistema.

En los siguientes diagramas se muestran los diferentes componentes que tiene cada funcionalidad que son: Interfaz de usuario, control y base de datos. Dependiendo de la funcionalidad que se esté definiendo aparecerán unos mensajes u otros. Los nombres de los componentes serán también diferentes.

- **Autenticar**

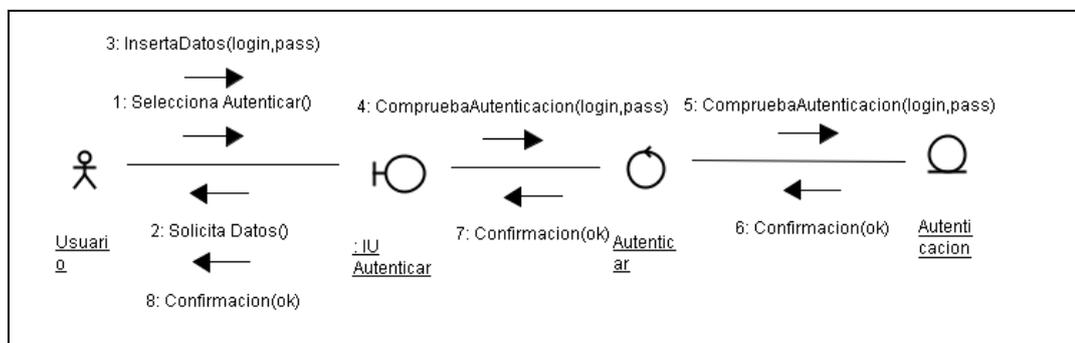


Figura 13: Diagrama de Colaboración. Autenticar

- **Solicitar Licencia**

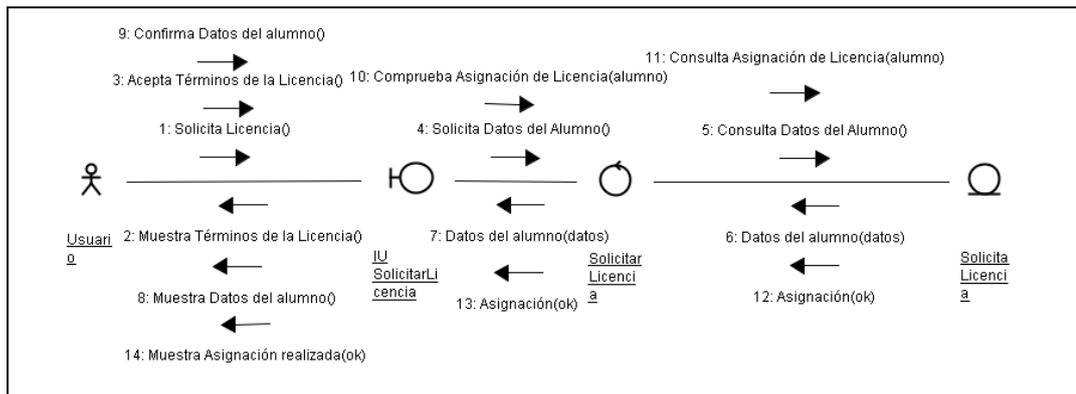


Figura 14: Diagrama de Colaboración. Solicitar Licencia

- **Descargar Software**

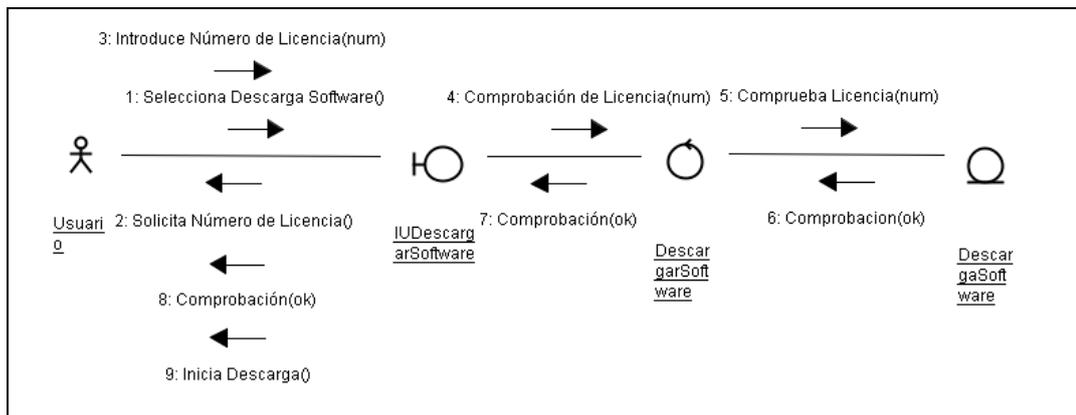


Figura 15: Diagrama de Colaboración. Descargar Software

- **Consultar Licencias**

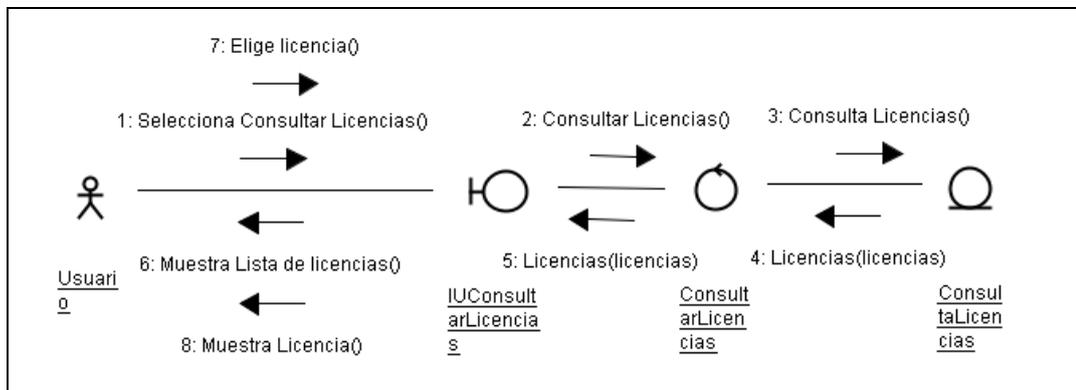


Figura 16: Diagrama de Colaboración. Consultar Licencias

- **Insertar Datos**

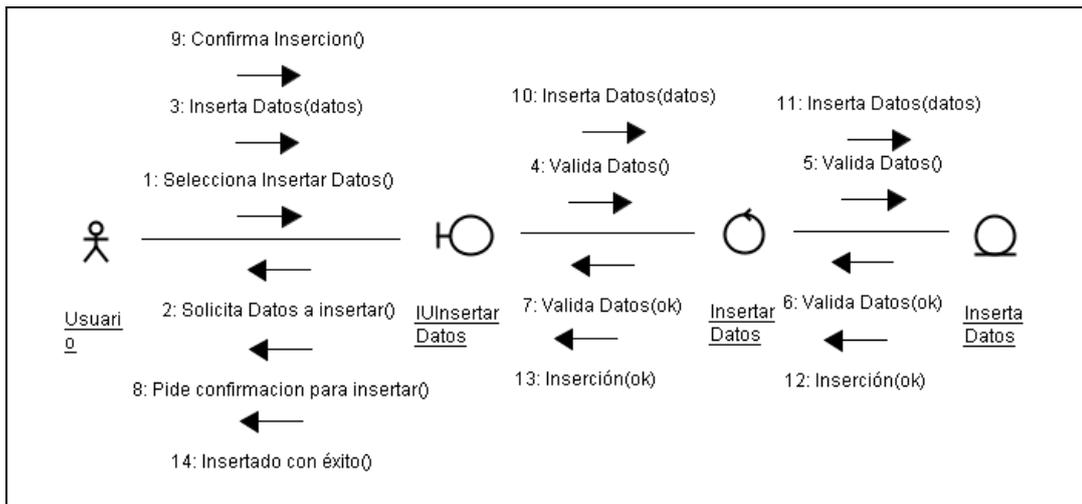


Figura 17: Diagrama de Colaboración. Insertar Datos

- **Modificar Datos**

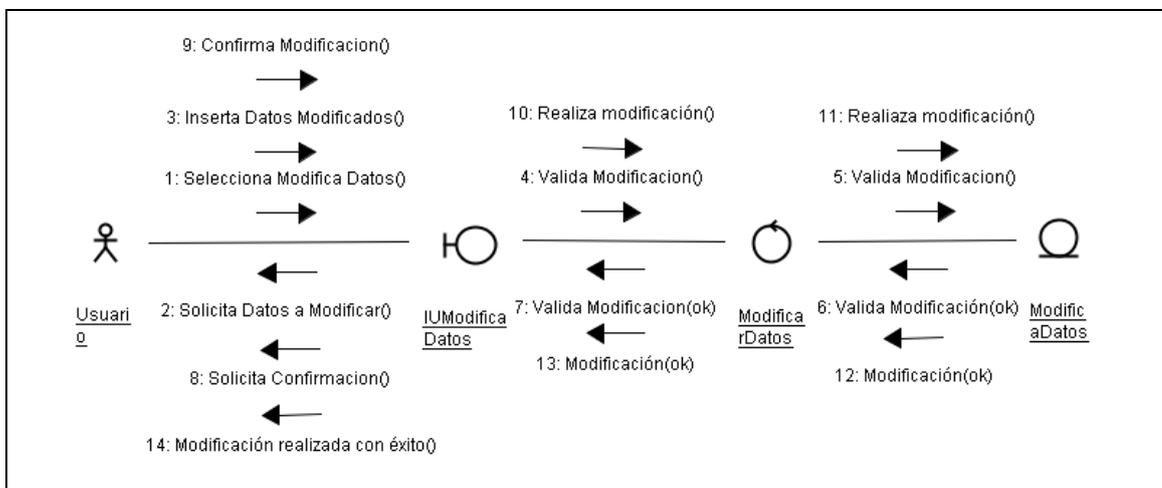


Figura 18: Diagrama de Colaboración. Modificar Datos

- **Eliminar Datos**

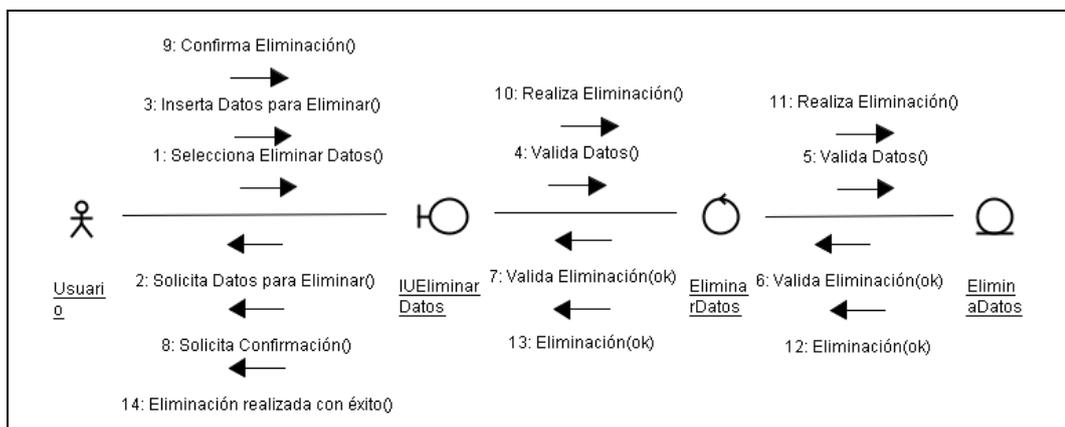


Figura 19: Diagrama de Colaboración. Eliminar Datos

- Consultar Datos

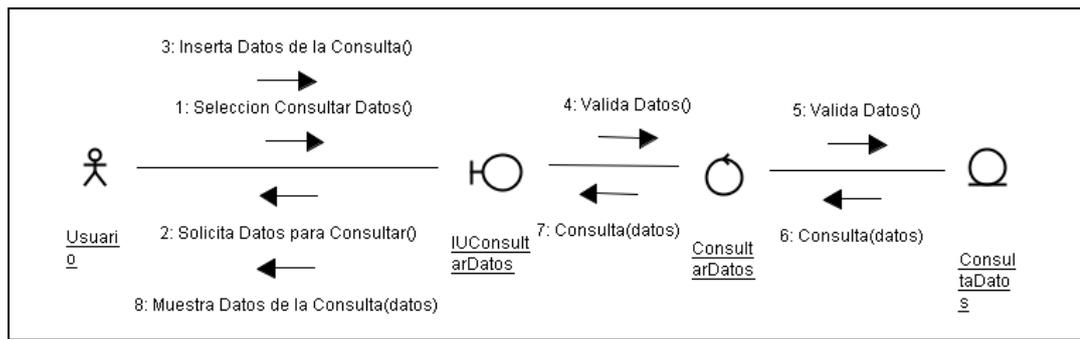


Figura 20: Diagrama de Colaboración. Consultar Datos

4.3. Diseño de la arquitectura del Sistema

Nuestra arquitectura está basada en el modelo de tres capas. La utilización de este modelo tiene como principal objetivo la separación de la lógica de negocios de la lógica de diseño. De este modo, diferenciaremos entre: la capa de presentación, negocio y datos.

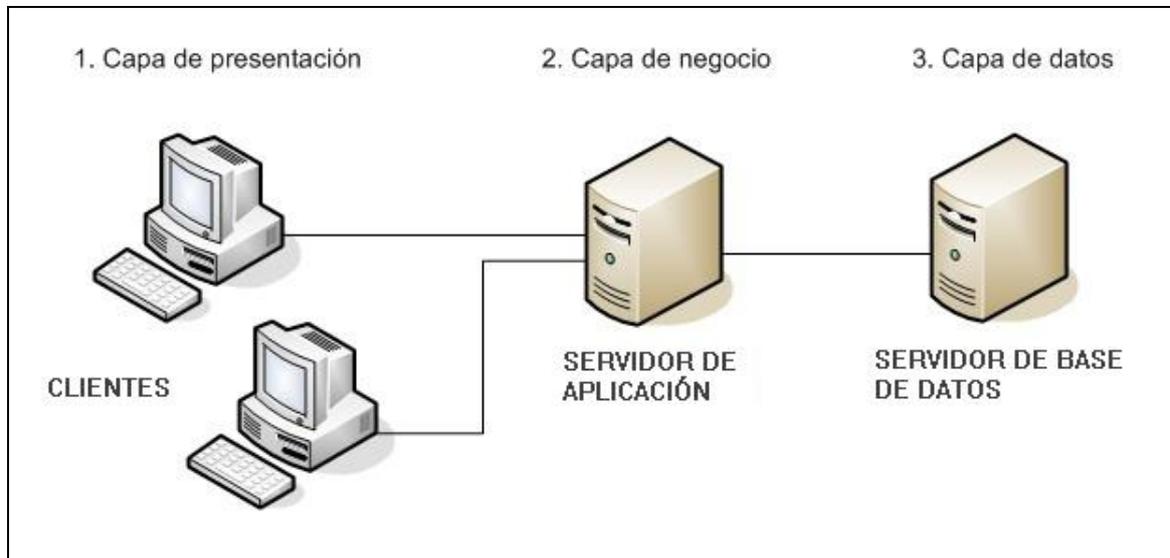


Figura 21: Modelo de tres capas

La capa de presentación hace las funciones de interfaz gráfica presentando el sistema al usuario. Esta capa contiene únicamente componentes de la IU con el fin de mostrar toda la información que necesite el usuario. Además, toma los datos de usuario necesarios para cada una de las operaciones que éste quiera realizar. Esta capa se comunica sólo con la capa de negocio.

La capa de negocio contiene todos los programas de nuestro sistema que implementan la lógica de negocio de la aplicación. Sirve de puente entre las otras dos capas. Por un lado, se comunica con la capa de presentación recibiendo las solicitudes y presentando los resultados. Por el otro lado, se comunica con la capa de datos para almacenar o recuperar datos que necesite en sus operaciones.

En la capa de datos residen los datos utilizados en nuestro sistema. Formada por un SGBD también se encarga de acceder a los mismos cuando la capa de negocio se lo solicita. Sólo se comunica con la capa de negocio.

4.4. Diseño de Clases

A continuación presentamos el diagrama de clases. Con este diagrama se puede apreciar la estructura del sistema, sus clases, atributos y las relaciones entre ellos.

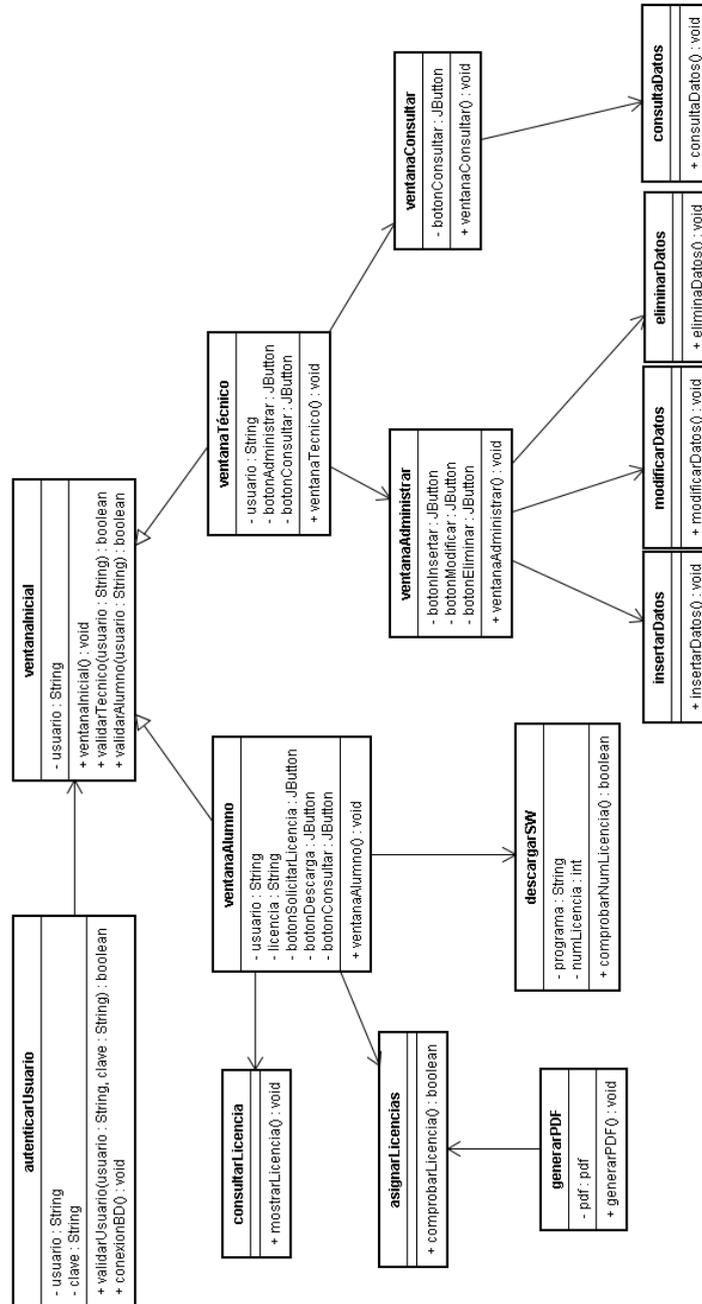


Figura 22: Diagrama de clases

4.5. Diseño de las Bases de Datos

En este apartado mostraremos el diseño de las Bases de Datos que utilizaremos en nuestra aplicación. Tendremos un B.D. que se ha realizado mediante el SGBD MySQL. El diseño de esta B.D. surge a partir de la especificación de requisitos que realizamos en el análisis. Además, usaremos otra B.D. basada en un SGBD Oracle para realizar la autenticación de usuarios.

Empezaremos hablando de la B.D. de MySQL. Esta B.D. precisa tener tablas que relacionen determinadas asignaturas online, que tendremos en la tabla “Rel_Asig_Soft”, con un software específico que almacenaremos en la tabla “Software”. Cada software tendrá una licencia asociada, por lo que tendremos que relacionar cada software de la tabla “Software” con una licencia de la tabla de “Licencias”. Cada vez que se asigne una licencia de la tabla de “Licencias” se tiene que llevar un control de a quién se le asigna, por eso creamos la tabla “Usuarios_Licencias”. Esta tabla contiene todas las asignaciones de licencias que se han realizado y a qué usuario se le han asignado. Por último, tendremos una tabla de administradores. Esta tabla contiene los usuarios que tienen permiso para entrar como técnico en la aplicación.

A continuación se muestra una imagen que contiene el diseño de la B.D.

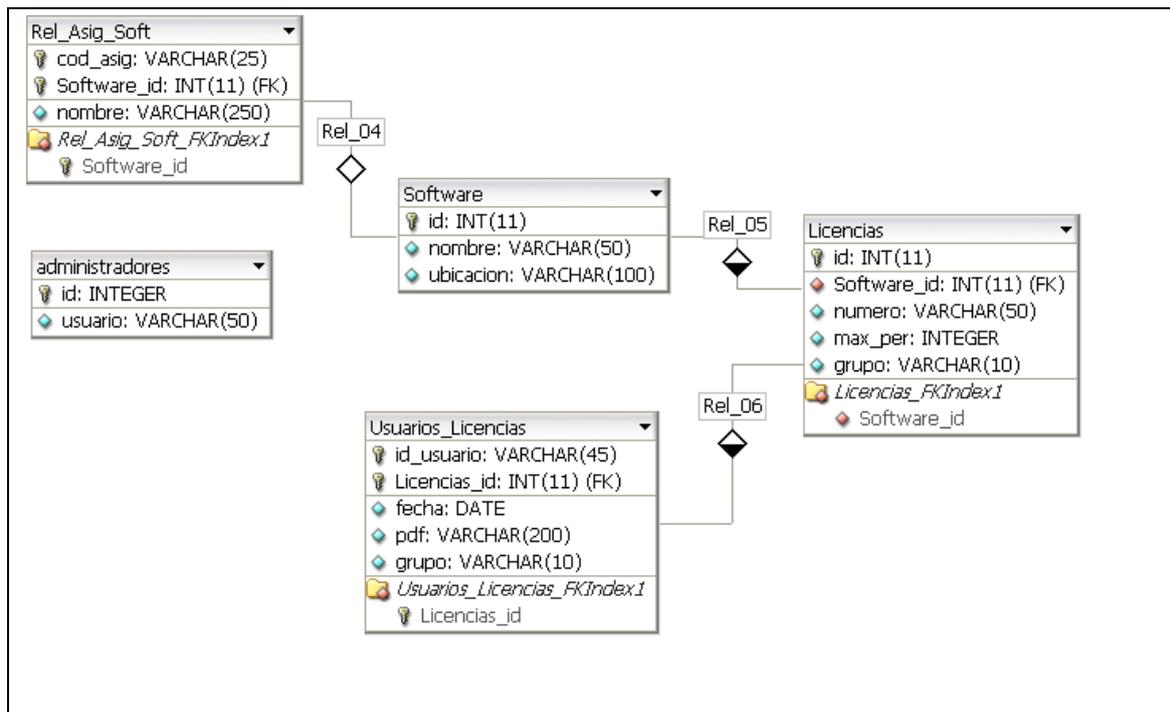
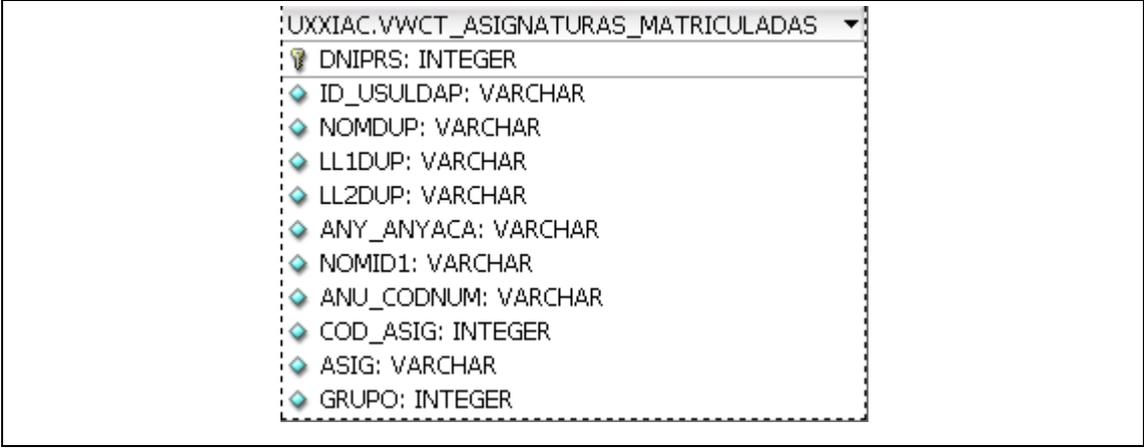


Figura 23: Diseño de la B.D. MySQL

La otra B.D. que utilizará nuestra aplicación es la que está controlada con el SGBD de Oracle. Utilizaremos esta B.D. para realizar la autenticación de los alumnos cuando quieren acceder a la aplicación. En la siguiente figura mostramos la tabla que usaremos en la autenticación.



The image shows a screenshot of a database tool displaying the structure of a table named 'UXXIAC.VWCT_ASIGNATURAS_MATRICULADAS'. The table has the following columns and data types:

Column Name	Data Type
DNIPRS	INTEGER
ID_USULDAP	VARCHAR
NOMDUP	VARCHAR
LL1DUP	VARCHAR
LL2DUP	VARCHAR
ANY_ANYACA	VARCHAR
NOMID1	VARCHAR
ANU_CODNUM	VARCHAR
COD_ASIG	INTEGER
ASIG	VARCHAR
GRUPO	INTEGER

Figura 24: Diseño de la B.D. Oracle

Comprobaremos que el alumno se encuentra en la B.D. y además cursa alguna de las asignaturas que requieren de la descarga de un software específico.

CAPÍTULO 5

5. Implementación y pruebas

5.1. Lenguaje de Programación

La aplicación AVGL utilizará principalmente PHP como lenguaje de programación. También se utilizarán algunas funciones JavaScript que nos permitirán mostrar mensajes al usuario en determinadas funcionalidades.

5.2. Detalles de implementación

En este apartado profundizaremos en los puntos más importantes de la aplicación. Explicaremos cómo se han implementado y se ilustrará con partes de código fuente.

Para empezar, una de las partes más importantes de la aplicación es la autenticación de los usuarios. Cuando un usuario se autentica, nos tenemos que asegurar que realmente es un usuario que tiene permiso para acceder al sistema. Por lo tanto, este punto es básico para la seguridad de AVGL. Una de las medidas utilizadas ha sido la creación de funciones que eviten vulnerabilidades SQL.

A continuación se puede ver parte del código utilizado:

```
<?php
// Con esta función subsanamos la vulnerabilidad de SQL Inyection
function parsearEntrada($entrada, $link){
    $parseo = mysql_real_escape_string($entrada, $link);
    return $parseo;
}
function parsearUsuario($entrada){
    $valido = false;
    if (preg_match('/^[a-z\d_\.,;]{0,30}$/i', $entrada)) {
        $valido = true;
    }
    return $valido;
}
?>
```

Código para controlar vulnerabilidades SQL

Como se dijo anteriormente tendremos dos tipos de usuarios, alumnos online y técnicos de la aplicación. Para autenticarse, usarán su usuario y contraseña de dominio único. Por lo tanto nuestra aplicación, se conectará al LDAP de la URJC para validar que los usuarios que se intentan conectar existen en LDAP. Además de esto, tendremos que conectarnos con la B.D. según quien sea el usuario que quiera acceder al sistema.

Si accede un alumno tendremos que conectarnos con la B.D. de Oracle de donde sacaremos las asignaturas en las que está matriculado. Después nos conectaremos a MySQL que es donde tenemos almacenados los códigos de las asignaturas que nos dan acceso a AVGL. Si alguna de las asignaturas en las que está matriculado el alumno, que hemos conseguido conectándonos en la B.D. de Oracle, coincide con alguna de las asignaturas almacenadas en MySQL, entonces la aplicación permitirá al alumno acceder al sistema. Si, por el contrario, quien accede a la aplicación es un técnico nos conectaremos sólo a la B.D. de MySQL. En esta B.D. se comprobará que el usuario tiene permisos de técnico y en caso afirmativo el usuario entrará en la aplicación. En las siguientes capturas mostraremos parte del código utilizado para realizar la autenticación.

Comenzaremos con la conexión a LDAP. En la llamada a la función que tenemos más abajo, las variables correo y clave recogen el usuario y contraseña de dominio único que el usuario ha insertado. Esta función se conectará a LDAP y comprobará que el usuario se encuentra ahí registrado.

```
$accederLDAP = ValidarUsuarioLDAP($correo, $clave);
```

Llamada a la función que valida usuarios de LDAP

A continuación mostraremos parte de la función ValidarUsuarioLDAP. En el extracto de código que aparece tenemos la parte donde se comprueba que haya alguna coincidencia entre el correo y contraseña que pasamos como parámetro y los que tenemos registrados en LDAP. Si esto ocurre, la función devuelve true.

```

//Conectarse al ldap
$ds = ldap_connect ('ldap.pruebas.urjc.es',389) or die ("No se puede conectar al ldap");
//echo "El resultado de la conexi&oacute;n es ".$ds."&lt;p>";
ldap_set_option($ds, LDAP_OPT_PROTOCOL_VERSION, 3);
//PARA INFODOC
$ldapbind = ldap_bind($ds, 'uid=campus_virtual,ou=admin,dc=urjc,dc=es', 'azE18bXa902');
$sr=ldap_search($ds,"ou=gente, dc=urjc, dc=es", "((&(Mail=".$correo.") (userPassword=".$userpassword.")))");
$info = ldap_get_entries($ds, $sr);

if ($info['count']==0){$conexion=false;
}else{$conexion=true;}
ldap_close ($ds);
return($conexion);

```

Extracto de código de la función ValidarUsuarioLDAP

Ahora mostraremos la función que comprueba si un usuario es técnico de la aplicación.

```

<?php
function validarAdmin($usuario, $link){
    $query = "select * from administradores where usuario like '$usuario'";
    $result = consultarMySQL($query, $link);
    $num_rows = mysql_num_rows($result);
    if($num_rows)
        $admin =true;
    else
        $admin = false;
    return $admin;
}
?>

```

Extracto de código de la función ValidarAdmin

Para terminar con la parte de autenticación mostraremos la parte del código donde, en el caso de que el usuario sea alumno online, comprobamos las asignaturas en las está matriculado.

```

//Query para sacar todas los codigos de asignaturas del usuario en la BD Oracle
$query = "select ANY_ANYACA, COD_ASIG, ASIG, GAC_CODNUM from
        UXXIAC.VWCT_ASIGNATURAS_MATRICULADAS
        where ID_USULDAP LIKE '$usuario'";
$s = consultarOracle($db_conn, $query);
while ($res = oci_fetch_array($s, OCI_ASSOC)) {
    $accedeOracle = true;
    // Construyo codigo de asignatura para cada fila
    $asig = $res['ANY_ANYACA'] . "-" . $res['COD_ASIG'] . "-" . $res['GAC_CODNUM'];
    $query2 = "select * from Rel_Asig_Soft where cod_asig like '$asig'";
    // Compruebo que este el codigo de asignatura dentro de la tabla Asignaturas_Acceso
    $result = consultarMySQL($query2, $link);
    $num_rows = mysql_num_rows($result);
    if($num_rows) {
        $row2 = mysql_fetch_array($result);
        $nombreAsig=$row2['nombre'];
        $anno = $res['ANY_ANYACA'];
        $accede = true;
        break;
    }
}
}

```

Extracto de código para comprobar las asignaturas del alumno

Otro aspecto a destacar en la implementación de AVGL es la creación de los PDF. Para este punto se utilizó una librería de libre distribución llamada FPDF. FPDF es una clase escrita en PHP que permite generar documentos PDF directamente desde PHP. La F de FPDF significa *Free* (gratis y libre), es decir, puede usarse para cualquier propósito y modificarla según nuestras necesidades. Además FPDF no necesita de ninguna extensión para PHP (excepto zlib para activar la compresión y GD para soporte a GIF) y funciona con PHP4 ($\geq 4.3.10$) y PHP5. Para crear los PDF tendremos un fichero PHP que importará la citada librería y contendrá las funciones necesarias para generar el archivo.

```
session_start();
require('./fpdf16/fpdf.php');
```

Extracto de código para importar la librería FPDF

A continuación, incluiremos las capturas de algunas de las funciones más importantes. Las funciones van formando secuencialmente el PDF. Tenemos funciones para la cabecera, el pie y el cuerpo del documento.

```
function Header()
{
    global $title;

    $this->Image('./imagenes/Campus Virtual Nuevo.PNG',10,8,33);
    $this->Ln(15);
    //Arial bold 15
    $this->SetFont('Arial','B',15);
    //Calculamos ancho y posición del título.
    $w=$this->GetStringWidth($title)+6;
    $this->SetX((210-$w)/2);
    //Colores de los bordes, fondo y texto
    $this->SetDrawColor(8,8,138);
    $this->SetFillColor(255,255,255);
    $this->SetTextColor(8,8,138);
    //Ancho del borde (1 mm)
    $this->SetLineWidth(1);
    //Título
    $this->Cell($w,9,$title,1,1,'C',true);
    //Salto de línea
    $this->Ln(10);
}
```

Extracto de código que genera la cabecera del PDF

```

function Footer()
{
    //Posición a 1,5 cm del final
    $this->SetY(-15);
    //Arial itálica 8
    $this->SetFont('Arial','I',8);
    //Color del texto en gris
    $this->SetTextColor(128);
    //Número de página
    $this->Cell(0,10,'Página '.$this->PageNo(),0,0,'C');
}

```

Extracto de código que genera el pie del PDF

```

function LicenseBody($file)
{
    //Leemos el fichero
    $f=fopen($file,'r');
    $txt=fread($f,filesize($file));
    fclose($f);
    //Times 12
    $this->SetFont('Times','',12);
    //Imprimimos el texto justificado
    $this->MultiCell(0,5,$txt);
    //Salto de línea
    $this->Ln();
}

```

Extracto de código que genera el cuerpo del PDF

Por último, vamos a incluir dos extractos de código. Uno que muestra cómo se escribe el texto en el documento y se insertan los datos y número de licencia asignado, y otro que nos enseña las líneas de código que crean y guardan el PDF generado en el servidor.

```

function PrintLicense($num,$file,$numLicen)
{
    $nombre = $_POST['nombre'];
    $apellidos = $_POST['apellidos'];
    $dni = $_POST['dni'];
    $correo = $_POST['correo'];

    $this->AddPage();
    $this->LicenseBody($file);
    $this->SetFont('Times','BU',14);
    $this->Cell(0,5,'DATOS PERSONALES DEL ALUMNO');
    $this->Ln(10);
    $this->DataUser('NOMBRE: ', $nombre);
    $this->DataUser('APELLIDOS: ', $apellidos);
    $this->DataUser('DNI: ', $dni);
    $this->DataUser('CORREO ALUMNO: ', $correo);
    $this->SetFont('Times','BU',14);
    $this->Ln(10);
    $this->Cell(0,5,'NÚMERO DE LICENCIA');
    $this->Ln(10);
    $this->DataUser('', $numLicen);
}

```

Extracto de código que escribe el texto del PDF

```
function createDir($dir)
{
    if (is_dir($dir))
        $this->Ln();
    else{
        //echo'Vamos a crear el fichero';
        mkdir($dir,0777);
    }
}
```

Extracto de código que guarda el PDF en el servidor

Existen otros muchos detalles de implementación. Pero la particular manera de autenticar usuarios y la generación de documentos PDF, son dos de los aspectos más destacables de AVGL.

5.3. Implementación de la arquitectura

En la etapa de diseño de la arquitectura de AVGL definimos una arquitectura de tres capas: presentación, negocio y datos. En este apartado, mediante un diagrama de componentes y otro de despliegue, explicaremos cómo se ha implementado la arquitectura de nuestra aplicación.

Primero presentaremos el diagrama de componentes. En él se pueden apreciar claramente las tres capas. El explorador Web será la capa de presentación, los tres componentes que aparecen en medio del diagrama forman la capa de negocio, mientras que los tres que tenemos más a la derecha serán la capa de datos. Cada caja representa los diferentes componentes de los que consta la aplicación.

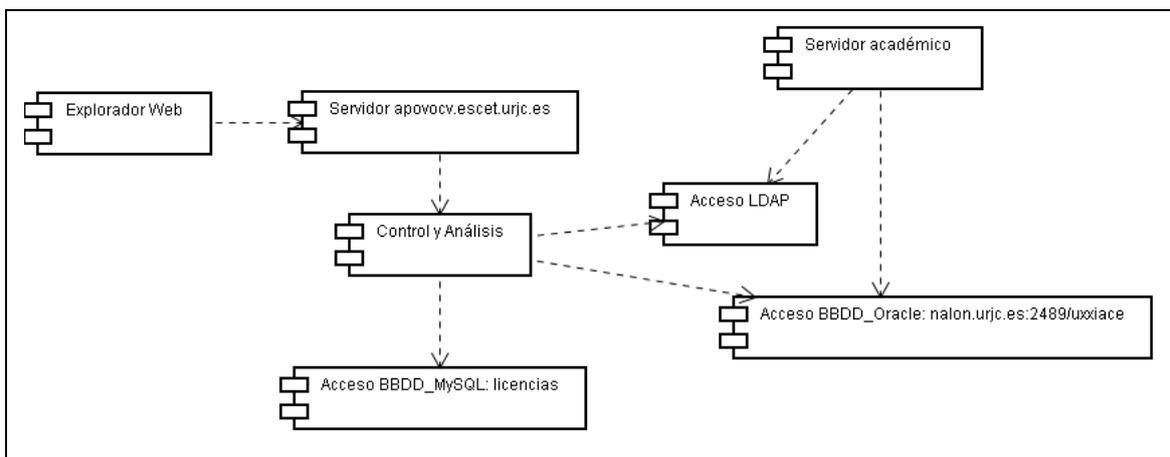


Figura 25: Diagrama de Componentes

En función del diagrama de componentes obtenemos el diagrama de despliegue. En él aparecerán tres nodos diferentes. Cada uno de ellos se corresponde con una de las tres capas de nuestra arquitectura.

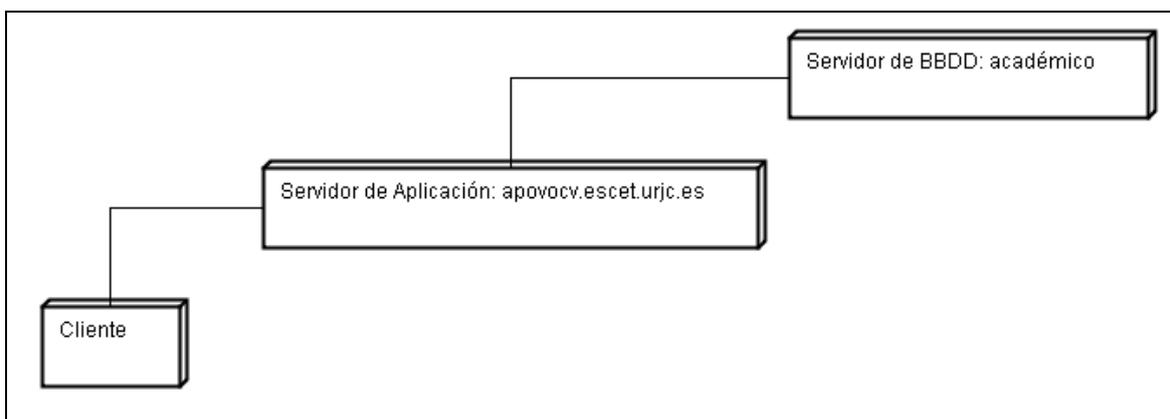


Figura 26: Diagrama de Despliegue

5.4. Estructura de directorios

La aplicación se encuentra alojada en el servidor *apoyocv.escet.urjc.es*. Se puede conectar a través de SSH.

La estructura de directorios de la aplicación se puede ver en la siguiente imagen.

AlumnosLicencias	Folder	22/02/2010 13:32:09	drwxrwxr...
config	Folder	22/02/2010 13:32:08	drwxrwxr-x
dir1	Folder	22/02/2010 13:32:08	drwxrwxr-x
estilos	Folder	16/03/2010 12:04:43	drwxrwxr-x
fpdf16	Folder	22/02/2010 13:32:08	drwxrwxr-x
Funciones	Folder	20/04/2010 11:10:27	drwxrwxr-x
general	Folder	22/02/2010 13:32:09	drwxrwxr-x
imagenes	Folder	17/03/2010 13:13:12	drwxrwxr-x
licencias	Folder	22/02/2010 13:32:09	drwxrwxr-x
programas	Folder	22/02/2010 13:34:56	drwxr-xr-x
Pruebas	Folder	16/03/2010 12:00:11	drwxrwxr-x
recursos	Folder	22/02/2010 13:32:03	drwxr-xr-x
sesiones	Folder	22/02/2010 13:32:09	drwxrwxr-x
autenticar.php	4,379 PHP Script	12/03/2010 10:05:37	-rw-r--r--
CompruebaNumLicenciaPS.php	1,279 PHP Script	22/02/2010 13:35:31	-rw-r--r--
CompruebaNumLicenciaQK....	1,279 PHP Script	22/02/2010 13:32:09	-rw-r--r--
descargaPS.php	6,494 PHP Script	22/02/2010 13:32:08	-rw-r--r--
descargaQK.php	7,096 PHP Script	22/02/2010 13:32:08	-rw-r--r--
formularioPS.php	3,952 PHP Script	22/02/2010 13:32:03	-rw-r--r--
formularioQK.php	3,948 PHP Script	22/02/2010 13:32:08	-rw-r--r--
index.php	4,962 PHP Script	20/04/2010 10:29:17	-rw-r--r--
licenciaPS.php	4,612 PHP Script	22/02/2010 13:32:03	-rw-r--r--
licenciaQK.php	4,752 PHP Script	22/02/2010 13:32:09	-rw-r--r--
login.php	7,019 PHP Script	20/04/2010 12:57:09	-rw-r--r--
pdfPS.php	3,848 PHP Script	21/04/2010 10:48:50	-rw-r--r--
pdfQK.php	4,060 PHP Script	21/04/2010 10:49:06	-rw-r--r--
principal.php	3,898 PHP Script	16/03/2010 14:08:08	-rw-r--r--
principalBIS.php	3,901 PHP Script	08/04/2010 16:55:07	-rw-r--r--
vistaFormularioPS.php	4,239 PHP Script	22/02/2010 13:32:09	-rw-r--r--
vistaFormularioQK.php	4,400 PHP Script	22/02/2010 13:32:08	-rw-r--r--
vistaPrincipal.php	6,794 PHP Script	20/04/2010 10:31:47	-rw-r--r--
vistaPrincipalAdmin.php	2,964 PHP Script	20/04/2010 12:45:23	-rw-r--r--
vistaPrincipalAdminBIS.php	2,778 PHP Script	08/04/2010 16:56:41	-rw-r--r--

Figura 27: Estructura de directorios

De todo el árbol de directorios que aparece en la figura vamos a comentar aquellos ficheros y carpetas que son más importantes.

En el directorio “AlumnosLicencias” es donde se irán almacenando los ficheros PDF de todas las licencias que se vayan asignando. Se organizará primero por el año académico,

después por el tipo de software y por último la asignatura a la que pertenecen los alumnos que han solicitado la licencia.

En la carpeta “Funciones” es donde tendremos los ficheros que implementan algunas de las funciones más importantes del sistema.

En “licencias” iremos guardando los textos con los términos y condiciones de las licencias. Si queremos cambiar algún texto bastaría con entrar en esta carpeta y cambiar el fichero correspondiente.

En el directorio “recursos” tendremos los diferentes programas que pueden descargarse los alumnos. Si queremos añadir algún programa nuevo o cambiar alguno de los que ya hay deberemos hacerlo en este directorio.

Por último, señalar que la mayoría de los ficheros con extensión “php” pertenecen a interfaces de la aplicación.

5.5. Pruebas

El sistema se ha validado mediante una serie de evaluaciones que detallaremos a lo largo de este apartado.

Dado que este proyecto ha seguido un proceso de desarrollo con funcionalidades bien definidas, a lo largo de su desarrollo se han realizado, siempre que fueran necesarias, pruebas para la verificación del correcto funcionamiento.

Se han realizado pruebas unitarias, utilizando técnicas de caja blanca, por cada funcionalidad creada en la aplicación. De esta manera, se han ido validando todas las partes y funcionalidades de manera individual. Posteriormente, se han desarrollado pruebas de integración. Se realiza este tipo de pruebas para conseguir obtener el sistema final uniendo las distintas partes individuales creadas. Por último, las pruebas de sistema, en las que simplemente se introduce una entrada y se obtiene una salida, sin conocer el funcionamiento interno (caja negra), son las que han validado el sistema de manera global y han probado que se realizaban las funcionalidades requeridas.

A continuación mostramos una tabla resumen de las pruebas realizadas a lo largo de la realización de este proyecto.

Prueba			Resultado
Validar Usuario	1. Validar Alumno		✓
	2. Validar Técnico		✓
Funcionalidades Alumno	3. Solicitar Licencia		✓
		3.1 Existen Licencias disponibles	✓
		3.2 No existen Licencias disponibles	✓
		3.3 No permitir descargar varias licencias para el mismo programa	✓
	4. Generar PDF		✓
	5. Comprobar Número de licencia		✓
	6. Descargar SW		✓
Funcionalidades Técnico	7. Consultar Datos		✓
		7.1 Existen Datos consultados	✓
		7.2 No existen Datos consultados	✓
	8. Insertar Datos		✓
		8.1 Introduce los datos correctamente	✓
		8.2 Introduce los datos incorrectamente	✓
	9. Modificar Datos		✓
		9.1 Existe registro	✓
		9.2 No existe registro	✓
		9.3 Introduce los datos correctamente	✓
		9.4 Introduce los datos incorrectamente	✓
	10. Eliminar Datos		✓
	10.1 Existe registro	✓	
	10.2 No existe registro	✓	

Tabla 1: Resumen de pruebas realizadas

5.6. Manual del usuario

Este manual va dirigido tanto a los alumnos como a los técnicos que tengan acceso a AVGL.

Lo hemos dividido en tres secciones:

- “Entrando a AVGL”.
- “Descargando Software”.
- “AVGL sección técnica”.

En la primera parte mostraremos los pasos que debe seguir un usuario para entrar a la aplicación. En el siguiente apartado del manual comenzaremos detallando las partes en las que se divide el menú principal de los alumnos. Después explicaremos paso a paso todo el proceso que se debe realizar para descargar un programa concreto.

Para probar esta parte de la aplicación existe un usuario de pruebas:

- **Usuario:** fran
- **Contraseña:** fran123

El último punto de este manual va dedicado al apartado técnico. Mediante capturas de pantalla y breves descripciones se irán explicando detalladamente todas las funcionalidades a las que tiene acceso un técnico. Para poder probar esta sección el usuario debe estar dado de alta como administrador en la B.D. de MySQL.

➤ ENTRANDO A AVGL

Index: Es la primera página que se encuentra un usuario al entrar a AVGL. Es una página informativa donde se explica la utilidad de nuestra aplicación y donde damos una serie de consejos y recomendaciones para usarla correctamente.

Aula Virtual URJC

Contáctenos | [Página principal](#)

Recursos

- Webmail
- Secretaría Virtual
- Gente
- Protección de Datos
- CAU
- Biblioteca universitaria

BIENVENIDO AL AULA VIRTUAL

¿Qué es?

El Aula Virtual es un servicio creado por el Campus Virtual que servirá de apoyo técnico para la descarga de software específicos con o sin licencia, los pasos que tienen que dar para descargarlos y ejecutarlos, y las normas y las condiciones legales para hacer uso de los mismos.

¿A quien va dirigido?

El Aula Virtual permite el acceso únicamente a los alumnos que estén matriculados de asignaturas en las que se requiera de un software específico para el cual necesiten algún tipo de licencia.

¿Que navegador recomendamos?

Para acceder se recomienda usar Mozilla Firefox. En caso de usar otro navegador es posible que la aplicación no funcione correctamente. En cualquier caso, para el funcionamiento correcto de la aplicación sugerimos que habilite el uso de las ventanas emergentes.

¿Problemas de acceso?

En caso de tener problemas de acceso al Aula Virtual, escribir un correo desde "Gestión Campus Virtual".

[Entrar en AULA VIRTUAL](#)

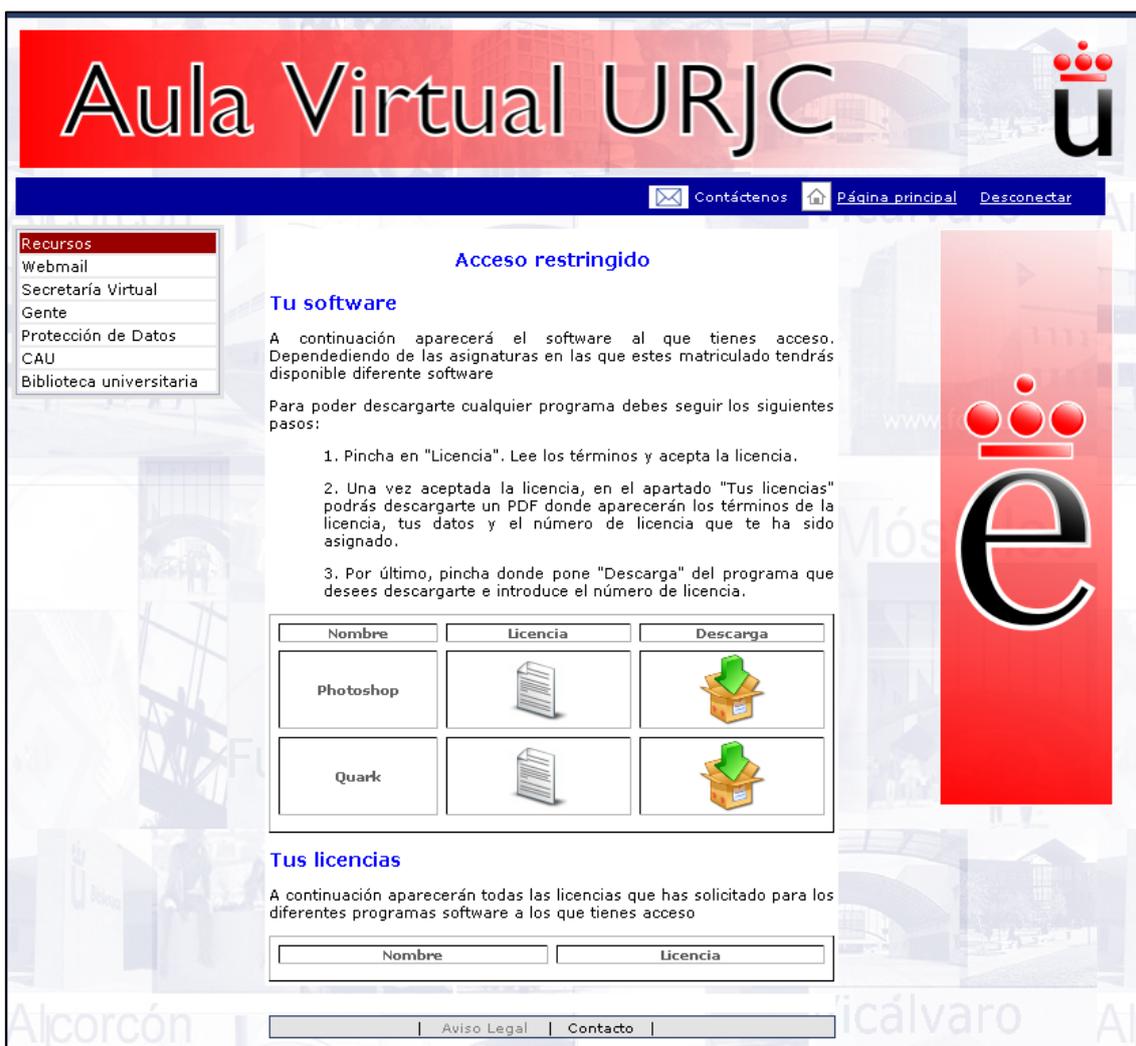
[Aviso Legal](#) | [Contacto](#)

Autenticar: Tras la pantalla index, lo primero que nos encontramos es la interfaz de autenticación. Como se puede ver en la siguiente captura, para poder acceder AVGL, el usuario simplemente tendrá que introducir su usuario y contraseña de dominio único, que es la misma que utiliza habitualmente para entrar en cualquier servicio de la universidad.



➤ DESCARGANDO SOFTWARE

Principal: Una vez el alumno se autentica correctamente, se encontrará con el menú que aparece en la siguiente imagen. Este menú consta de dos secciones “Tu software” y “Tus licencias”. Cada alumno, dependiendo de las asignaturas en las que esté matriculado, tendrá diferente software para descargar. En el caso del ejemplo nuestro alumno puede descargarse los programas de Photoshop y Quark.



The screenshot shows the 'Aula Virtual URJC' interface. At the top, there is a red banner with the text 'Aula Virtual URJC' and the URJC logo. Below the banner is a navigation bar with links for 'Contáctenos', 'Página principal', and 'Desconectar'. On the left side, there is a 'Recursos' menu with options like 'Webmail', 'Secretaría Virtual', 'Gente', 'Protección de Datos', 'CAU', and 'Biblioteca universitaria'. The main content area is titled 'Acceso restringido' and 'Tu software'. It contains instructions on how to download software and a table listing available software for download.

Nombre	Licencia	Descarga
Photoshop		
Quark		

Below the table, there is a section for 'Tus licencias' with a table for listing licenses:

Nombre	Licencia
<input type="text"/>	<input type="text"/>

Los pasos para descargar cualquier software aparecen en el menú de la imagen. Estos serían:

1. Pincha en el icono "Licencia". Lee los términos y acepta la licencia.

2. Una vez aceptada la licencia, en el apartado "Tus licencias" aparecerá un PDF en el cual aparecerán los términos de la licencia, tus datos personales y el número de licencia que te ha sido asignado. Te recomendamos que te lo descargues, aunque siempre podrás tener acceso a una copia en este mismo apartado.

3. Por último, pincha en el icono "Descarga" del programa que desees descargarte e introduce el número de licencia que te hemos indicado anteriormente.

Ahora, a través de capturas y breves instrucciones iremos explicando cada uno de estos pasos detalladamente.

✓ PASO 1:

Licencia: Tras pinchar en “Licencia” accederemos a una página donde aparecerán los requisitos legales y normas de uso de nuestro programa. El alumno deberá aceptar dichos requisitos. En la siguiente imagen se muestra un ejemplo de las condiciones de una licencia de Photoshop.

Protección de Datos
CAU
Biblioteca universitaria

Requisitos legales y normas de uso de software Photoshop CS

- El alumno dispondrá de un número de licencia del software de Photoshop cuyo titular es la Universidad Rey Juan Carlos.
- Dicha licencia se presta únicamente al alumno de la Universidad Rey Juan Carlos que esté matriculado en cualquiera de las siguientes asignaturas: "Tecnologías Digitales I: Edición y Producción" - GRUPO ÚNICO con código 2009-10_1009128_94320, de 4º curso de la titulación de Licenciatura en Periodismo On Line, y "Tecnologías Multimedia" - GRUPO ÚNICO, con código 2009-10_1009421_94290, de 3º curso de la titulación en Publicidad On Line.
- Dicha licencia por lo tanto sólo estará disponible para el curso académico 2009-2010 y bajo las condiciones que marca el Campus Virtual de la Universidad Rey Juan Carlos.
- Dicha licencia da derecho a utilizarla única y exclusivamente para los fines educativos que marque el profesor de dicha asignatura.
- El alumno sólo podrá hacer uso de la licencia en un único ordenador, que será donde realice las prácticas habitualmente para cursar la carrera de Periodismo On Line, y estará obligado a desinstalar el programa de su ordenador en Julio de 2010.
- El uso y/o el manejo indebido de dicha licencia con otros fines que no sean educativos, como por ejemplo fines lucrativos, la distribución del programa y de su licencia, o el incumplimiento de las normas anteriormente expuestas, es única y exclusivamente responsabilidad del alumno, quien se tendrá que enfrentar a las acciones legales pertinentes.
- Una vez el alumno rellene dicho documento y nos lo envíe a través del espacio de "Gestión Campus Virtual" , el Campus Virtual le enviará el número de licencia personal en un documento adjunto que necesitará posteriormente cuando instale el software.
- El alumno se hace responsable de la salvaguarda del número de la licencia y de su manejo personal e intransferible.
- La Universidad Rey Juan Carlos se guarda el derecho de anular, modificar o eliminar dicha licencia al alumno que no cumpla los requisitos legales y las normas de uso anteriormente citadas.

Los datos personales recogidos serán incorporados y tratados en el fichero informatizado "CAMPUS-VIRTUAL", cuya finalidad es el control administrativo y de gestión de los alumnos y profesores del Campus Virtual. Este fichero se encuentra en proceso de inscripción en el Registro de Ficheros de Datos Personales de la Agencia de Protección de Datos de la Comunidad de Madrid (www.madrid.org/apdcm), y no podrán ser cedidos, salvo cesiones previstas en la Ley.

- El órgano responsable del fichero es Campus Virtual-URJC, y la dirección donde el interesado podrá ejercer los derechos de acceso, rectificación, cancelación y oposición ante el mismo es campusvirtual@urjc.es, todo lo cual se informa en cumplimiento del artículo 5 de la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.
- El alumno, mediante el presente documento completo, acepta su consentimiento y se compromete a cumplir las normas anteriormente descritas.

Acepto Atrás

Datos Personales: Una vez que el alumno acepta los requisitos legales y normas de uso, el sistema le mostrará sus datos personales extraídos de la B.D. de académico. Esta información la utilizaremos para conformar el futuro PDF, el cual nos servirá como contrato de licencia. Estos datos no podrán ser modificados por el alumno, así que si alguno está mal el alumno tendrá que ponerse en contacto con campusvirtual@urjc.es para subsanarlo.



The screenshot shows the 'Aula Virtual URJC' interface. At the top, there is a navigation bar with 'Contáctenos', 'Página principal', and 'Desconectar'. A sidebar on the left lists 'Recursos' such as 'Webmail', 'Secretaría Virtual', 'Gente', 'Protección de Datos', 'CAU', and 'Biblioteca universitaria'. The main content area is titled 'Acceso restringido' and 'Datos personales'. It contains a warning message: 'A continuación aparecerán tus datos personales. Estos datos han sido obtenidos de nuestra Base de Datos académica. Antes de enviar comprueba que todos tus datos personales son correctos.' Below this is a form with the following fields:

DATOS PERSONALES	
Nombre	Francisco
Apellidos	Escalero
DNI	123456789A
Correo de alumno	fran@alumnos.urjc.es
<input type="button" value="Enviar"/> <input type="button" value="Cancelar"/>	

At the bottom of the form area, it says: 'Si alguno de tus datos no es correcto ponte en contacto con nosotros: campusvirtual@urjc.es'. The footer contains 'Aviso Legal' and 'Contacto' links.

Envío de Datos Personales: El alumno, al pulsar en “Enviar”, confirma que sus datos personales son los correctos. Como se puede ver en la imagen, sale una ventana emergente para confirmar el envío. Cuando confirmamos, nuestro sistema genera automáticamente un PDF que nos servirá como contrato de licencia entre el alumno y la universidad.

The screenshot displays the 'Aula Virtual URJC' interface. At the top, there is a navigation bar with links for 'Contáctenos', 'Página principal', and 'Desconectar'. A sidebar on the left lists 'Recursos' such as 'Webmail', 'Secretaría Virtual', 'Gente', 'Protección de Datos', 'CAU', and 'Biblioteca universitaria'. The main content area is titled 'Acceso restringido' and 'Datos personales'. It contains a form with the following fields: 'DNI' (123456789A) and 'Correo de alumno' (fran@alumnos.urjc.es). A confirmation dialog box is overlaid on the form, asking '¿Estas seguro de enviar este formulario?' with 'Aceptar' and 'Cancelar' buttons. Below the form, there is a note: 'Si alguno de tus datos no es correcto ponte en contacto con nosotros: campusvirtual@urjc.es'. At the bottom, there are links for 'Aviso Legal' and 'Contacto'. The URJC logo is visible in the top right corner, and a large 'e' logo is on the right side of the page.

✓ PASO 2:

Guardar Licencia: Tras el envío, volveremos al menú principal y una ventana emergente nos sugiere que guardemos en nuestro ordenador el PDF. Este PDF estará disponible en el menú del usuario siempre que quiera consultarlo o descargarlo. Este documento es importante, ya que en él, aparecerán entre otros datos, el número de licencia que le ha sido asignado.

The screenshot shows the 'Aula Virtual URJC' website. At the top, there is a navigation bar with links for 'Contáctenos', 'Página principal', and 'Desconectar'. A sidebar on the left lists 'Recursos' such as 'Webmail', 'Secretaría Virtual', 'Gente', 'Protección de Datos', 'CAU', and 'Biblioteca universitaria'. The main content area is titled 'Acceso restringido' and 'Tu software'. It contains a warning dialog box that says: 'La página en http://pruebascv.escet.urjc.es ... IMPORTANTE: En el siguiente PDF aparecerá su número de licencia. Guarde el PDF en su ordenador. Aceptar'. Below the dialog, there is a table with columns 'Nombre', 'Licencia', and 'Descarga' showing software like Photoshop and Quark. Further down, there is a section 'Tus licencias' with a table showing the license for Photoshop. At the bottom, there are links for 'Aviso Legal' and 'Contacto'.

Nombre	Licencia	Descarga
Photoshop		
Quark		

Nombre	Licencia
Photoshop	

Licencia: A continuación mostramos un ejemplo con un extracto del PDF que se genera. Si tenemos activado el uso de ventanas emergentes, el navegador nos abrirá una nueva pestaña que nos permitirá ver la licencia completa. Esta es la misma licencia que tendrá disponible el alumno en cualquier momento para consultar o descargar desde su apartado “Tus licencias”.



Requisitos legales y normas de uso de software Photoshop CS

DATOS PERSONALES DEL ALUMNO

NOMBRE: Francisco

APELLIDOS: Escalero

DNI: 123456789A

CORREO ALUMNO: fran@alumnos.urjc.es

NÚMERO DE LICENCIA

NÚMERO DE LICENCIA: 33333

✓ PASO 3:

Principal: Como podemos ver en la siguiente captura, ahora en el menú del alumno dentro del apartado "Tus licencias" aparece un PDF con la licencia asignada. Como ya hemos dicho, el alumno podrá acceder a ella cuando quiera.

The screenshot displays the 'Aula Virtual URJC' interface. At the top, there is a navigation bar with links for 'Contáctenos', 'Página principal', and 'Desconectar'. A sidebar on the left lists various resources like 'Webmail', 'Secretaría Virtual', and 'Biblioteca universitaria'. The main content area is titled 'Acceso restringido' and 'Tu software'. It provides instructions on how to download software and lists two programs: Photoshop and Quark. Below this, the 'Tus licencias' section shows a table with a PDF icon for the Photoshop license. A large red vertical banner with a crown and the letter 'e' is visible on the right side of the interface.

Nombre	Licencia	Descarga
Photoshop		
Quark		

Nombre	Licencia
Photoshop	

Introducir número de licencia: Una vez aceptados los términos de la licencia, y con el número de licencia asignado, el alumno podrá descargarse el programa. Para ello, pinchando sobre el icono de descarga que teníamos en el menú, se le abrirá una página nueva donde aparecerán, además de los requisitos del sistema, un link para iniciar la descarga del software.



The screenshot shows a web page with a blue header containing navigation links: 'Contáctenos', 'Página principal', and 'Desconectar'. On the left, there is a 'Recursos' menu with items like 'Webmail', 'Secretaría Virtual', 'Gente', 'Protección de Datos', 'CAU', and 'Biblioteca universitaria'. The main content area is titled 'Acceso restringido' and 'Requisitos del programa'. It explains that minimum PC requirements are needed for Photoshop and lists them for Windows and Macintosh. A 'DESCARGAR PHOTOSHOP' button is present, along with instructions for users without a license and a link to a manual. A large red vertical banner with a crown and the letter 'e' is on the right. The footer contains 'Aviso Legal', 'Contacto', and the URL 'www.fcjs.urjc.es'.

Acceso restringido

Requisitos del programa

Para poder ejecutar dicho software son necesarios unos mínimos requisitos técnicos del PC donde se instale Photoshop:

WINDOWS

- Intel Pentium 4, Intel Centrino, Intel Xeon o Intel™ Core Duo (o compatible)
- Microsoft Windows XP con Service Pack 2 o Windows Vista™ Home Premium, Business, Ultimate o Enterprise (certificado para ediciones de 32 bits)
- 512 MB de RAM
- 64 MB de RAM de vídeo
- 1,5 GB de espacio disponible en el disco duro (se necesita espacio libre adicional durante la instalación)
- Resolución de pantalla de 1.024 x 768 con tarjeta de vídeo de 16 bits

MACINTOSH

- PowerPC Procesador Intel G4, G5 o multinúcleo o Intel
- Mac OS X v10.4.8-10.5 (Leopard)
- 512 MB de RAM
- 64 MB de RAM de vídeo
- 1,5 GB de espacio disponible en el disco duro (se necesita espacio libre adicional durante la instalación)
- Resolución de pantalla de 1.024 x 768 con tarjeta de vídeo de 16 bits

Para que puedas instalar el programa de Photoshop:

- Necesitas disponer de conexión de banda ancha en el ordenador donde se vaya a descargar el programa (independientemente del ordenador donde se vaya a instalar), ya que la descarga del programa ocupa 250 MB.
- A través del siguiente enlace podrás descargarte el programa:

DESCARGAR PHOTOSHOP
(Al pulsar espere unos segundos)

Si aún no tiene número de licencia pinche [aquí](#)

Si tuvieras algún tipo de dificultad o de duda sobre cómo instalarte el programa, puedes descargarte un manual explicativo:
- [Manual de Instalación de Photoshop CS](#)

[Aviso Legal](#) | [Contacto](#) | [www.fcjs.urjc.es](#)

Descargar Software: Al pinchar sobre el link de descargar aparecerá una ventana emergente que nos pedirá el número de licencia. Si éste es correcto, la aplicación permitirá iniciar la descarga de software.

The screenshot shows a web browser window with a blue header containing navigation links: "Contáctenos", "Página principal", and "Desconectar". On the left, a "Recursos" menu lists: "Webmail", "Secretaría Virtual", "Gente", "Protección de Datos", "CAU", and "Biblioteca universitaria". The main content area is titled "Acceso restringido" and "Requisitos del programa". It states that to run the software, minimum PC requirements are needed. Under "WINDOWS", the requirements are:

- Intel Pentium 4, Intel Centrino, Intel Xeon o Intel™ Core Duo (o compatible)
- Microsoft Windows XP con Service Pack 2 o Windows Vista™ Home Premium, Business, Ultimate o Enterprise (certificado para

A "Mozilla Firefox" dialog box is open, titled "Número de Licencia". It contains a text input field with the placeholder "Introduzca aquí su número de licencia" and "Número de licencia:" to its left. Below the input are "Aceptar" and "Cancelar" buttons. A link "aquí" is provided for users without a license number. The dialog box status bar shows "Terminado".

Below the dialog, there is a "DESCARGAR PHOTOSHOP" button with the instruction "(Al pulsar espere unos segundos)". Another "aquí" link is provided for users without a license number. A link to a "Manual de Instalación de Photoshop CS" is also present. The footer contains "Aviso Legal" and "Contacto" links, and the URL "www.fcjs.urjc.es".

Ventana de descarga: A continuación, aparece una ventana que, si todo ha ido bien, confirmará que nuestro número de licencia es correcto. Después, simplemente pinchando en la opción deseada se iniciará la descarga.



Ventana de descarga: Al final, después de haber pinchado en uno de los links disponibles para la descargar del software, tendremos esta ventana que nos permite guardar el software en nuestro ordenador.



➤ AVGL SECCIÓN TÉCNICA

Principal: Este es el menú que se encuentra el administrador al entrar. Hay dos secciones “Consulta” y “Administrar B.D.”.

The screenshot shows the administrator interface for the Aula Virtual URJC. At the top, there is a red banner with the text "Aula Virtual URJC" and the URJC logo. Below the banner is a navigation bar with links for "Contáctenos", "Página principal", and "Desconectar". On the left side, there is a menu titled "Recursos URJC" with options: "Webmail", "Secretaría Virtual", "Gente", "Protección de Datos", "CAU", and "Biblioteca universitaria". The main content area is titled "Acceso restringido" and "Cuenta de Administrador". It contains a message: "A continuación aparecerá el menú de administrador. Desde esta cuenta podrá gestionar y consultar el Aula Virtual." Below this message is a red bar with two tabs: "Consulta" and "Administrar BD". Under the "Administrar BD" tab, there are two dropdown menus: "Año: 2009-10" and "Software: PHOTOSHOP". Below these menus is a "Consultar" button. On the right side, there is a large red vertical banner with the URJC logo and a large stylized "e". At the bottom of the page, there is a footer with links for "Aviso Legal" and "Contacto".

Consulta: En esta captura se muestra un ejemplo de consulta. En este caso, se han sacado todos los alumnos de todas las asignaturas del año 2009-2010 que usaron Photoshop. Se puede ver y descargar cualquier licencia de cualquier alumno. También tendrá la opción de descargarse todas las licencias en formato “zip”.



Administrar B.D.: En esta página aparecerán todas las tablas que hay en la B.D. de MySQL. Simplemente elegimos que tabla que queremos editar.

Aula Virtual URJC

Contáctenos | Página principal | Desconectar

Recursos URJC
Webmail
Secretaría Virtual
Gente
Protección de Datos
CAU
Biblioteca universitaria

Acceso restringido

Cuenta de Administrador

A continuación aparecerá el menú de administrador. Desde esta cuenta podrá gestionar y consultar el Aula Virtual.

Consulta | **Administrar BD**

Elija la tabla que quiera administrar:

Nombre	Tabla
Licencias	
Rel_Asig_Soft	
Software	
Usuarios_Licencias	
administradores	

Aviso Legal | Contacto

Administrar Tabla: Este es un ejemplo donde queremos administrar la tabla de administradores. Una vez elegimos la tabla se mostrarán todos los registro que contiene. Ahora podemos elegir entre insertar un nuevo registro, editar un registro existente o borrar un registro.



The screenshot shows the 'Aula Virtual URJC' administrator interface. At the top, there is a red banner with the text 'Aula Virtual URJC' and the URJC logo. Below the banner, there is a navigation bar with links for 'Contáctenos', 'Página principal', and 'Desconectar'. On the left side, there is a menu titled 'Recursos URJC' with items like 'Webmail', 'Secretaría Virtual', 'Gente', 'Protección de Datos', 'CAU', and 'Biblioteca universitaria'. The main content area is titled 'Acceso restringido' and 'Cuenta de Administrador'. It contains a message: 'A continuación aparecerá el menú de administrador. Desde esta cuenta podrá gestionar y consultar el Aula Virtual.' Below this, there are two tabs: 'Administrar BD' (selected) and 'Consulta'. The 'Administrar BD' tab shows a table titled 'Tabla: administradores' with the following data:

id	usuario
1	g.contreras
2	alejandro.galindo
3	santiago.moral
4	javier.ramos.decastro
5	a.galindo
6	natalia.esteban
7	y.pasichnyk

Below the table, there are three buttons: 'Añadir registro', 'Editar registro', and 'Borrar registro', along with an 'Atrás' button. At the bottom of the page, there are links for 'Aviso Legal' and 'Contacto'. On the right side, there is a large red vertical banner with the URJC logo and a stylized 'e'.

Insertar Registro: Al pulsar sobre la opción añadir registro saldrá un pequeño formulario donde aparecerán los campos del nuevo registro que queremos insertar. Sencillamente los rellenamos y pulsamos en “Aceptar”.

Aula Virtual URJC

Contáctenos | Página principal | Desconectar

Recursos URJC

- Webmail
- Secretaría Virtual
- Gente
- Protección de Datos
- CAU
- Biblioteca universitaria

Acceso restringido

Cuenta de Administrador

A continuación aparecerá el menú de administrador. Desde esta cuenta podrá gestionar y consultar el Aula Virtual.

Administrador BD | Consulta

Tabla: administradores

id	usuario
1	g.contreras
2	alejandro.galindo
3	santiago.moral
4	javier.ramos.decastro
5	a.galindo
6	natalia.esteban
7	v.pasichnyk

Insertar registro

id:

usuario:

Aceptar | Reset

Añadir registro | Editar registro | Borrar registro

Atrás

Aviso Legal | Contacto

Editar Registro: Si pulsamos sobre editar registro aparecerá un formulario con los registros que son clave primaria de la tabla. Insertamos en los campos los datos del registro que queremos editar y pulsamos en Aceptar.

Aula Virtual URJC

Contáctenos | Página principal | Desconectar

Recursos URJC
Webmail
Secretaría Virtual
Gente
Protección de Datos
CAU
Biblioteca universitaria

Acceso restringido

Cuenta de Administrador

A continuación aparecerá el menú de administrador. Desde esta cuenta podrá gestionar y consultar el Aula Virtual.

Administrador BD | Consulta

Tabla: administradores

id	usuario
1	g.contreras
2	alejandro.galindo
3	santiago.moral
4	javier.ramos.decastro
5	a.galindo
6	natalia.esteban
7	y.pasichnyk

Editar registro

id:

Aceptar | Reset

Añadir registro | Editar registro | Borrar registro | Atrás

Aviso Legal | Contacto

Editar el registro: Tras insertar los datos del registro que queremos editar nos aparecerá otro formulario con todos los campos. Modificamos los campos que queramos y pulsamos en “Aceptar”.

Aula Virtual URJC

Contáctenos | Página principal | Desconectar

Recursos URJC

- Webmail
- Secretaría Virtual
- Gente
- Protección de Datos
- CAU
- Biblioteca universitaria

Acceso restringido

Cuenta de Administrador

A continuación aparecerá el menú de administrador. Desde esta cuenta podrá gestionar y consultar el Aula Virtual.

Administrar BD | Consulta

Tabla: administradores

id	usuario
1	g.contreras
2	alejandro.galindo
3	santiago.moral
4	javier.ramos.decastro
5	a.galindo
6	natalia.esteban
7	v.pasichnyk

Editar registro

id:

usuario:

Aviso Legal | [Contacto](#)

Borrar el registro: Al pinchar en borrar registro aparecerá un formulario donde se insertarán los datos del registro que se quiera borrar. Tras eso simplemente pulsamos en “Aceptar”.

Aula Virtual URJC

Contáctenos | Página principal | Desconectar

Recursos URJC
Webmail
Secretaría Virtual
Gente
Protección de Datos
CAU
Biblioteca universitaria

Acceso restringido

Cuenta de Administrador

A continuación aparecerá el menú de administrador. Desde esta cuenta podrá gestionar y consultar el Aula Virtual.

Administrar BD | Consulta

Tabla: administradores

id	usuario
1	g.contreras
2	alejandro.galindo
3	santiago.moral
4	javier.ramos.decastro
5	a.galindo
6	natalia.esteban
7	y.pasichnyk

Borrar registro

id:

Aceptar | Reset

Añadir registro | Editar registro | Borrar registro | Atrás

Aviso Legal | Contacto

6. Conclusiones

6.1. Trabajos futuros y mejoras

La aplicación funciona correctamente, y realiza bien todas las funciones que se le piden. Actualmente se encuentra en explotación y da servicio tanto a alumnos como a técnicos. Aún así se sigue trabajando para poder desarrollar trabajos futuros que amplíen su funcionalidad y hagan de AVGL una aplicación más potente.

Uno de los trabajos futuros pensados para la aplicación es ampliar la oferta de software descargable desde AVGL. De momento están operativos dos programas para que los alumnos se los puedan descargar, como son: Photoshop y QuarkXPress. La idea sería crear una aplicación de descargas, con todo tipo de programas, sean propietarios (es decir, software que necesite licencia) o no.

Otro posible trabajo futuro es ampliar el número de usuarios que pueden tener acceso a AVGL. Actualmente, aparte de los técnicos, los únicos usuarios que tienen acceso a AVGL son alumnos que están en modalidad online y, dentro de éstos, sólo pueden acceder aquellos que estén matriculados en determinadas asignaturas. Por lo tanto, un posible trabajo de futuro sería ampliar el acceso a otro tipo de usuarios como profesores, alumnos presenciales, etc.

En cuanto al capítulo de mejoras, existen algunos aspectos, principalmente en cuanto a implementación, que se pueden mejorar. A continuación pasamos a detallar algunos posibles cambios que aumentarían las capacidades de AVGL.

La aplicación es poco escalable. Como ya hemos dicho, anteriormente en esta versión de AVGL sólo tenemos la opción de descargarnos dos programas: Photoshop y Quark. Si en el futuro quisiéramos añadir algún programa más para descargar no sería complicado, pero si el número de programas aumenta considerablemente, el trabajo se multiplica. Para evitar esto tendríamos que cambiar el diseño de las páginas. En vez de tener páginas distintas para cada programa deberíamos de utilizar una “plantilla”. Es decir usar siempre la misma página, pero en función del software que queremos la página mostrase diferente contenido.

Otro aspecto mejorable es el de modularidad. El código debería de tener más funciones que evitarían mucha repetición de líneas de código.

En cuanto a las funcionalidades que tiene el técnico, existen algunos aspectos que se podrían mejorar. Por ejemplo, la aplicación podría gestionar la edición y borrado de registros simplemente con un botón de selección en vez de tener que insertar los datos del registro que queremos editar o borrar.

6.2. Estimación de esfuerzos

El presente proyecto se ha realizado en cuatro meses de trabajo. A la semana se han dedicado cinco días y cada día alrededor de cinco horas. Por lo tanto, la realización total del proyecto ha durado aproximadamente unas 400 horas.

A continuación, adjuntamos un cuadro resumen donde se desglosa el tiempo aproximado que se ha dedicado a cada una de las fases de nuestro proyecto.

Fase	Horas	%
Análisis	80	20%
Diseño	80	20%
Implementación	180	45%
Redacción	60	15%
Total	400	100%

Tabla 2: Cuadro resumen de la estimación de esfuerzos

En la fase de análisis se incluyen también los estudios previos realizados. La realización de esta fase ocupa entorno al 20% del tiempo total dedicado para realizar la aplicación. Al igual que para la fase de análisis en la fase de diseño se ha empleado entorno al 20% de las horas totales. La fase que más tiempo ha durado ha sido la que corresponde a la implementación de AVGL. Esta fase ha durado casi la mitad del tiempo. Por último, la redacción del presente documento ha durado entorno a 60 horas. Esto corresponde al 15% del tiempo total.

Bibliografía

LIBROS REFERENCIADOS

G. BOOCH, I. JACOBSON, J. RUMBAUGH, "EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE", EDITORIAL ADDISON WESLEY, 2005.

GRADY BOOCH, JAMES RUMBAUGH, IVAR JACOBSON, "EL LENGUAJE UNIFICADO DE MODELADO", EDITORIAL ADDISON WESLEY, 2000.

I.SOMMERVILLE, "INGENIERÍA DEL SOFTWARE", EDITORIAL ADDISON WESLEY, 2002.

PÁGINAS WEB REFERENCIADAS

EL PROCESO UNIFICADO:

- <http://epcc.unex.es/modules.php?op=modload&name=Downloads&file=index&req=descargar&lid=1920>
- <http://www.kybele.etsii.urjc.es/docencia/IS5/>
- http://es.wikipedia.org/wiki/Proceso_Unificado

PHP:

- <http://es.wikipedia.org/wiki/PHP>
- www.maestrosdelweb.com
- www.desarrolloweb.com

ORACLE:

- <http://es.wikipedia.org/wiki/Oracle>
- <http://www.oracle.com/global/lad/corporate/story.html>

MYSQL:

- <http://dev.mysql.com/>

LDAP:

- <http://es.wikipedia.org/wiki/LDAP>
- <http://www.ldap-es.org/contenido/04/11/1.2.-ventajas-en-el-uso-de-ldap>
- <http://delta.cs.cinvestav.mx/~fraga/Charlas/ldap.PDF>

LIBRERÍA DE PDF:

- <http://www.fPDF.org/>