



PFC APLICACIONES WEB PARA TRABAJO COLABORATIVO
DESARROLLO DE UN BACKOFFICE DE GESTIÓN
COLABORATIVO PARA UNA APLICACIÓN WEB DE
RESERVAS ONLINE

ALUMNO: RAÚL ROS PEREZ
TUTOR: FATOS XHAFA

Índice PFC

1. Resumen ejecutivo.....	3
2. Introducción.....	4
2.1. Razón y oportunidad del proyecto.....	4
2.2. Situación actual.....	4
3. Descripción del proyecto.....	5
3.1. Objetivos del proyecto.....	5
3.2. Resultados esperados.....	6
3.3. Análisis de riesgos.....	6
4. Alcance de la propuesta.....	7
4.1 Desarrollo de software de gestión de clientes y facturación.....	7
4.2 Elementos de gestión colaborativa para una mayor organización y comunicación.....	7
4.3 Evaluación de sistema mediante herramientas de test estrés.....	7
4.4 Elaboración de entregables del proyecto.....	7
5. Plan de trabajo.....	8
5.1. Relación de actividades.....	8
5.1.1 Lanzamiento del Proyecto.....	8
5.1.2 Especificación de Requisitos.....	8
5.1.2 Análisis.....	13
5.1.2.1 Casos de Uso Resumidos.....	13
5.1.2.2 Casos de Uso Detallados.....	15
5.1.3 Diseño.....	28
5.1.3.1 Diseño de Interfaces.....	28
5.1.3.2 Diseño de Diagrama de Clases.....	41
5.1.3.3 Diseño del Modelo Físico de la Base de datos.....	43
5.1.4 Implementación.....	45
5.1.4.1 Control de Perfiles.....	45
5.1.4.2 Área colaborativa.....	49
5.1.4.3 Área de Chat.....	51
5.1.4.4 Área de administración de usuarios.....	54
5.1.5 Instalación y configuración.....	58
5.1.6 Pruebas de estabilidad y carga.....	61
5.1.7 Algunas pantallas de la aplicación en funcionamiento.....	68
5.2. Hitos principales del proyecto.....	70
5.3. Calendario de trabajo.....	71
6. Organización del proyecto.....	73
6.1. Equipo de trabajo.....	73
6.2. Definición de roles.....	73
6.3. Mecanismos de control.....	73
7. Valoración Económica.....	73
8. Bibliografía.....	74

1. Resumen ejecutivo.

El Proyecto Final de Carrera (PFC) está pensado para realizar un trabajo de síntesis de los conocimientos adquiridos en varias asignaturas de la carrera y que requiera ponerlas en práctica conjuntamente en un trabajo concreto.

En el proyecto actual se ha abordado un problema existente en una empresa real de ahora en adelante llamada “empresa Z”. La empresa Z dispone de un *software* de reservas distribuido en modo *SaaS*(*Software as a Service*) que permite a los profesionales que trabajan con cita previa como por ejemplo fisioterapeutas, médicos, *spas*, peluqueros, centros de negocio, ... la posibilidad de gestionar todas sus citas a través de un aplicativo *web*. Cuando un profesional utiliza la herramienta obtiene una serie de ventajas, entre otras, no se necesita instalar el *software* en el ordenador o la posibilidad de usar un módulo de cita previa en la página del profesional simplemente copiando y pegando un código *html* proporcionado por el *software*.

La empresa Z factura a los clientes de forma manual, activando sus cuentas en la base de datos y creando la facturas con un programa de ofimática. A medida que el *software* crece en usuarios, a la empresa le está siendo más complicado gestionar toda la facturación que emite la aplicación. Para solucionar este problema, en el presente proyecto se ha creado un módulo para la aplicación que permite acceder a una gestión en tiempo real de los clientes.

Por otro lado, también se necesitaba gestionar dos tipos de usuarios que hasta ahora no se habían utilizado. colaboradores y partners trabajan en conjunto para que se puedan crear nuevas cuentas de clientes y obtener un sistema de ventas piramidal que permita a la “empresa Z” incrementar los beneficios y extenderse de una forma más rápida por el mercado.

Para controlar a estos partners y colaboradores se ha creado otro módulo en el que se establece un control de creación y alta colaborativo que permite a los administradores de la empresa Z controlar tanto a clientes, partners como colaborador.

Finalmente para poder solucionar incidencias y realizar configuraciones en cada una de las cuentas de los clientes, se ha desarrollado un nuevo módulo en que se ha habilitado un acceso para que tanto colaboradores, partners como administración pueda acceder a cada una de las instancias de las cuentas de los clientes, permitiendo así dar un soporte en tiempo real a cada uno de ellos.

El desarrollo del proyecto se ha realizado con tecnología PHP 5, Mysql 5 y Apache 2.2. El uso de esta tecnología viene establecido porque la aplicación de citación previa está creada en un framework propietario de la “empresa Z”.

2. Introducción.

La “empresa Z” dispone de un software para gestionar cita previa y distribuido en modo SaaS que tiene una serie de necesidades por completar. Estas necesidades están directamente relacionadas con las áreas comercial, facturación y de soporte. A continuación se realizará una descripción de la situación actual del proyecto y en que áreas del software se trabajará para terminar de completarlo y dotar a la empresa de una infraestructura de gestión más completa que la que posee ahora.

2.1. Razón y oportunidad del proyecto.

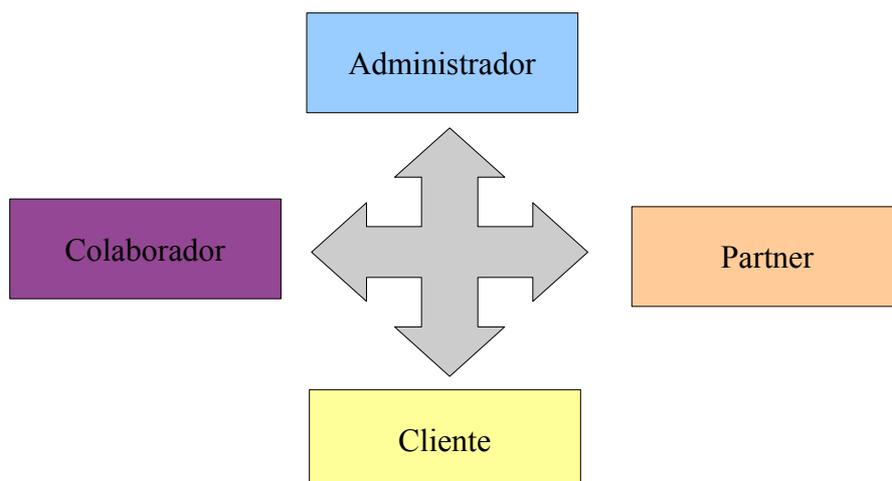
Nuestra empresa ha desarrollado un *software* de reservas online que tiene como ventajas entre otras, la integración sencilla de un widget de reserva online en la web de un cliente que quiera utilizar el sistema, no necesita instalación en un ordenador ya que el *software* está en la nube, dispone de una api para integración con otros sistemas y la utilización de un calendario con un *look & feel* estilo Google Calendar que da al usuario la ventaja de no tener que familiarizarse con la interfaz.

2.2. Situación actual.

En los últimos meses el éxito de este software está creciendo de forma rápida ya que empresas de cupones, multinacionales y empresas pequeñas se están interesando por su funcionamiento.

Gracias a este éxito, la compañía cree que es necesario iniciar el desarrollo de un módulo o módulos que complementen el software de citación y permitan la gestión de facturación de clientes, gestión de colaboradores, partners y el acceso a diferentes instancias para resolución en tiempo real de incidencias de configuración o pequeños problemas que puedan tener clientes que llamen por teléfono.

Lo que se pretende en el proyecto es además de crear una herramienta de gestión, crear una herramienta colaboración entre los diferentes actores del sistema. Los actores administradores, colaboradores, partners y clientes, actuarán en determinados casos de forma complementaria para que la información llegue correctamente a cada uno de estos.



En el siguiente punto del proyecto especificaremos con más detalle el proyecto y las formas de colaboración entre los diferentes actores del sistema.

3. Descripción del proyecto.

El proyecto actual consiste en la construcción de una herramienta de gestión colaborativa entre administradores, colaboradores, partners y clientes. Este software será construida como un módulo o módulos adicionales al software de gestión de reservas y cita previa que existe actualmente para permitir a cada uno de los diferentes actores una gestión y colaboración más óptima y organizada.

3.1. Objetivos del proyecto.

Al sistema se deberá acceder introduciendo un email y password. Dependiendo del tipo de actor o perfil que tenga cada usuario accederá a un panel o a otro.

Entre los diferentes actores que podemos encontrar están: administradores, colaboradores, partners, clientes.

Los **administradores** del sistema podrán realizar las siguientes tareas: gestionar clientes, gestionar colaboradores y partners, gestionar facturas e interactuar con la gestión colaborativa.

Un administrador es un empleado de la empresa Z y por ello debe tener el máximo control del sistema de gestión.

Los **colaboradores** del sistema podrán realizar las siguientes tareas: gestionar clientes, gestionar facturas e interactuar con la gestión colaborativa. Los colaboradores no podrán realizar determinadas tareas en la gestión de clientes o de facturas que sólo podrán ser realizadas por los administradores.

Los **partners** del sistema podrán realizar las siguientes tareas: gestionar clientes, gestionar facturas e interactuar con la gestión colaborativa.

Hay una serie de diferencias entre colaboradores y partners. Desde el punto de vista de negocio un partner trabajará con un contrato en el que se tiene que comprometer a una serie de objetivos, por ejemplo, alcanzar un número de ventas, dar soporte, contratos de exclusividad, etc. Además los partners tienen o pueden conseguir sus propios clientes con lo que ellos son los que facturan al cliente (con su propio software) y no la empresa Z. También pueden tener más libertad a la hora de acordar o cerrar los precios con los clientes. De este modo el sistema generará facturas a la empresa partner y no a cada uno de los clientes. Por otro lado los clientes conseguidos por los colaboradores si que son facturados por la empresa Z, el colaborador únicamente se encarga de vender el producto y llevarse una comisión.

Los **clientes** del sistema podrán descargar facturas e interactuar con el sistema colaborativo.

En la definición y análisis del proyecto se explicará de forma más detallada todas y cada una de las acciones que podrán realizar los usuarios.

3.2. Resultados esperados.

El software se desarrollará en un entorno web, es decir, podrá ejecutarse a través de un navegador y no necesitará instalación de ningún software adicional. El software tendrá que estar desarrollado en tecnología PHP 5, base de datos mysql 5 y servidor web apache 2.2.

Para conseguir que la ejecución del proyecto tenga éxito se deberán completar una serie de hitos intermedios.

1. Especificación de Requerimientos.
2. Análisis y diseño.
3. Implementación de las diferentes partes del software.
4. Instalación en un sistema uoc/no uoc para demostración.
5. Evaluación mediante herramientas de test estrés.

Finalmente se entregará el documento plan de trabajo completado con todas las secciones del proyecto.

En las secciones 5.2 y 5.3 se definirá de forma más precisa las fechas de entrega y la especificación de cada uno de los hitos de forma más detallada.

3.3. Análisis de riesgos.

Tras analizar el documento actual se han encontrado los siguientes riesgos.

1. Teniendo en cuenta las fechas de entrega es posible que no se disponga de tiempo suficiente para implementar todas las funcionalidades. Si esto ocurriera, se intentaría dar prioridad a los desarrollos relacionados con la gestión colaborativa.
2. Para implementar la tecnología chat se utilizarán librerías externas que faciliten su desarrollo ya que el tiempo es muy ajustado para la envergadura del proyecto y el desarrollo de un chat completo podría ralentizar el proyecto.

4. Alcance de la propuesta.

En el presente proyecto se considera:

1. Desarrollo de software de gestión de clientes y facturación.
2. Desarrollo de gestión colaborativa para una mayor organización y comunicación.
3. Evaluación de sistema mediante herramientas de test estrés.
4. Elaboración de entregables del proyecto.

4.1 Desarrollo de software de gestión de clientes y facturación.

En el proyecto se solicita la construcción de un software utilizando el framework de desarrollo PHP Nuvol. Este framework ha sido desarrollado íntegramente por la empresa Z. Para la construcción del software y para que sea complementario con el software existente actualmente se utilizará esta tecnología. Habrá que desarrollar todas las opciones explicadas en la especificación de requerimientos del proyecto.

4.2 Elementos de gestión colaborativa para una mayor organización y comunicación.

Además el software deberá incorporar un área colaborativa donde los diferentes actores del sistema podrán interactuar entre sí. El flujo de comunicación permitirá tanto comunicación en tiempo real a través de un chat como la comunicación a través de mensajes, emails y comunicaciones internas que integrará el sistema.

4.3 Evaluación de sistema mediante herramientas de test estrés.

Estamos hablando de un software que tiene que soportar cargas puntuales de estrés en peticiones simultáneas de usuarios. Se tendrá que realizar un pequeño análisis de la respuesta del sistema ante situaciones de sobrecarga. Para ello se utilizarán herramientas de comparativa que sometan al sistema a situaciones de estrés.

4.4 Elaboración de entregables del proyecto.

El proyecto tendrá que tener una serie de entregables que serán suministrados a la “Empresa Z”. Estos entregables estarán basados en UML e incluirán: especificación de requerimientos, documentos de análisis como casos de uso, resumido y detallado, documentos de diseño, mockups, diseño de base de datos, diagrama de clases y explicación de algoritmos de relevancia en el proyecto.

5. Plan de trabajo.

5.1. Relación de actividades.

En el punto actual se iniciará una explicación detallada del proyecto.

5.1.1 Lanzamiento del Proyecto.

El proyecto se inicia el día 5 de Marzo de 2012. Para ello se realiza una reunión interna en la “empresa Z” donde se obtiene la siguiente especificación de requisitos. Más adelante se ofrece un desglose de toda la planificación de tareas que se llevarán a cabo en el proyecto.

5.1.2 Especificación de Requisitos.

En esta sección se intenta dar una visión general de lo que debería realizar el sistema. Principalmente se dividirá en dos secciones, en la primera se explicará de forma general como funciona el sistema y lo que puede hacer o no cada uno de los actores. En la segunda, se hará especial hincapié a los diferentes métodos colaborativos que incorporará la herramienta entre los diferentes usuarios.

Explicación general.

Como ya se ha visto anteriormente, la empresa Z necesita la construcción de un software para gestionar su área interna de administración. Hemos hablado anteriormente de la necesidad de intervención de 4 tipos de usuarios, **administradores** que serán los empleados de la empresa Z, **colaboradores** que normalmente serán comerciales o representantes interesados en vender el software, **partners** que serán otras empresas comercializadoras de software o otras empresas que tengan un software y quieran integrarlo, etc y **clientes** que son los profesionales o empresas finales que utilizan el sistema de reservas para gestionarse y recibir citación online.

Además también se necesita la posibilidad de gestionar la facturación de una forma más automatizada y directa, evitando el acceso a la manipulación de la base de datos y la utilización de un software de ofimática para la elaboración de la facturación.

Por otro lado también es necesario la posibilidad de acceder a cada una de las instancias de las cuentas de los clientes debido a que en ocasiones es posible que los clientes necesiten ayuda en la configuración de la herramienta o resolver una duda. Para ello un partner o administrador deberá poder acceder a la instancia de la cuenta del cliente. Un colaborador no es necesario que acceda ya que este soporte lo ofrecerán o los partners o la empresa Z.

Como comentábamos, la aplicación podrá ser utilizada por 4 tipos de usuarios: Administradores, colaborador, Partners y Clientes.

Un **administrador** podrá gestionar clientes. La gestión de clientes consiste en ver un listado de clientes, poder buscar entre ellos por varios campos (nombre, email, país, teléfono, móvil), añadir clientes, eliminar clientes y actualizar clientes.

Un cliente es normalmente una empresa que quiere utilizar el motor de reservas para dar el servicio a sus usuarios de cita online. Se compone de los siguientes datos nombre, apellidos, email, país, provincia, ciudad, zona horaria, teléfono, teléfono móvil, dirección, número de mensajes sms disponibles, cif, número máximo de agendas, producto contratado, código de cupón, días de versión trial, fecha de caducidad contrato actual.

Además el administrador podrá activar/desactivar de una forma sencilla la cuenta de un cliente. Cuando se desactiva una cuenta no se elimina, simplemente queda en un estado “no utilizable” por el cliente.

Desde la sección de gestión de clientes también se permitirá acceder a cada una de las instancias de las cuentas de clientes. Esto es necesario debido a que los clientes suelen llamar por teléfono cuando tienen dudas de configuración o hay algún problema con la herramienta.

Un usuario **administrador** también podrá gestionar facturas a un cliente. El sistema de reservas permite la contratación directa del servicio a través de la web. El sistema permite pago mediante tarjeta, el cual hace el cobro de forma inmediata o transferencia. En el caso de transferencia, el cliente tiene que hacer un pago posterior indicando un determinado concepto. Lo que mostrará concretamente esta sección será un listado de las facturas que han sido pagadas o que están pendientes de pago.

A este área se podrá acceder de dos formas: directamente desde el menú general a un listado global de todas las facturas de todos los clientes ordenadas por fecha de emisión o a partir del listado de clientes a través de un acceso que filtre las facturas de ese cliente.

En ambos casos se establecerá un código de colores para una identificación más sencilla de las facturas que están cobradas y facturas que están pendientes de pago. Además también se podrán crear y emitir nuevas facturas a los clientes, actualizar facturas y eliminar facturas. Las facturas deberán contener los datos del cliente (normalmente una empresa) nombre, cif, dirección, ciudad, país, localidad. Las facturas podrán ser emitidas con IVA o sin IVA ya que en ocasiones los clientes suelen ser de países distintos a España o regiones a las que no se cobra iva. También deberán disponer la posibilidad de realizar un descuento.

Por otro lado, en el listado también se podrá buscar por diferentes campos y generar facturas en pdf. Tanto en facturas como clientes se podrá exportar el listado en formato csv.

El usuario **administrador** también podrá gestionar a los colaboradores y partners. colaborador y partner se compone de los siguientes datos, nombre de empresa (no obligado), nombre y apellidos de contacto, dirección, teléfono, teléfono móvil, código postal, país, ciudad, region, documento o cif (no obligado). El administrador podrá ver un listado de colaboradores y partners que se diferenciarán por el tipo. En este listado se podrán realizar búsquedas por teléfono, teléfono móvil, país, ciudad, region, por nombre de empresa o nombre de contacto y tipo. También se podrá exportar el listado en formato csv y podrá crear, modificar o eliminar partners y colaborador.

En el caso de los **colaborador** podrán ver a sus clientes y sus facturas. En cuanto a gestión de clientes podrán exportar un listado de sus clientes y no podrá ver clientes de otros colaboradores o partners. Además podrán realizar búsquedas por varios campos (nombre, email, país, teléfono, móvil). También podrán exportar el listado en formato csv.

Cuando activen/desactiven cuentas a sus clientes o creen/modifiquen/eliminen clientes necesitarán la aprobación de los administradores. En cualquier caso sí que podrán crear un tipo de cliente y cuenta especial *trial*, que no necesitará de la aprobación de los administradores. Las cuentas trial tienen un periodo de validez estándar de 30 días. Los colaboradores podrán dar hasta 90 días y aunque se notificará a los administradores, no será necesaria su aprobación ya que se necesita fluidez a la hora de ofertar este tipo de cuentas.

En cuanto a la gestión de facturas un **colaborador** podrá ver un listado de facturas de sus clientes de forma global ordenadas por proximidad de cobro y utilizando el mismo código de colores que los administradores. También podrán obtener las facturas de cada uno de sus clientes accediendo

a través de los datos de un cliente. Podrá buscar facturas por diversos campos, obtener un pdf y exportar el listado de facturas a csv. Además podrá visualizar por cada cliente que porcentaje de beneficio está ganando.

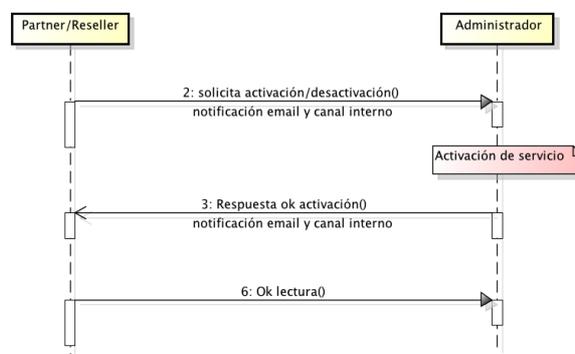
Un **partner** podrá realizar las mismas acciones que un colaborador pero con dos diferencias. Un partner podrá acceder a las instancias de aplicaciones que sus clientes tengan contratadas para realizar asistencias telefónicas para configuración y resolución de dudas. Las facturas irán emitidas a la empresa partner en lugar de a los clientes finales. Finalmente un cliente podrá ver un listado de las facturas que le han sido emitidas y podrá obtener las facturas en formato pdf.

Gestión colaborativa.

En el aspecto colaborativo entre los diferentes actores, se crearán una serie de canales de comunicación. Tanto **administradores**, como **colaboradores y partners** dispondrán de dos líneas de comunicación internas una de chat y una de mensajes. La línea de chat se utilizará para una comunicación directa y en tiempo real, la línea de mensajes se utilizará para realizar una comunicación menos directa y para utilizarla como flujo de aprobaciones. Esta última consistirá en un timeline en modo lista con las comunicaciones de cada uno de los colaboradores, partners y clientes pendientes de ejecutar algún tipo de acción, aprobación o rechazo. De este modo todos los actores podrán comunicarse con los administradores de una forma directa sin utilizar teléfono o email.

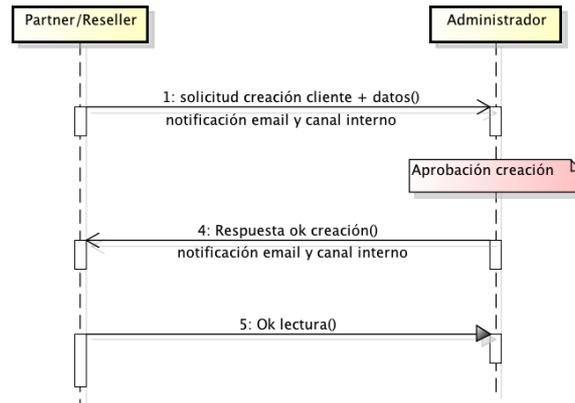
El chat puede ser utilizado para resolver dudas puntuales en el canal de venta o técnico. La línea de interna mensajes se utilizará para los siguientes casos:

- Un partner o colaborador puede solicitar la activación/desactivación de una cuenta de un cliente.



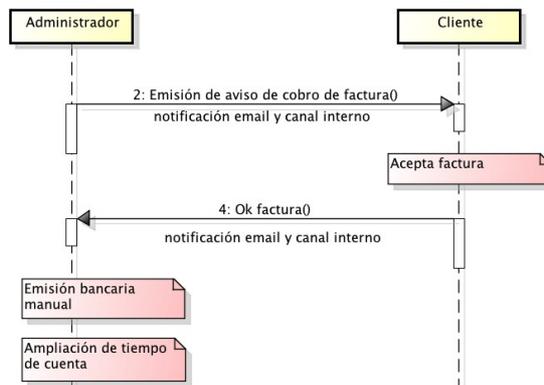
Cuando la petición llegue al administrador tendrá que aprobarla o rechazarla, si la aprueba se enviará una notificación al partner o colaborador, a su vez cuando el partner o colaborador lea la notificación del canal interno se enviará una confirmación de la lectura. En caso de rechazarla se enviará una notificación al partner/colaborador con el motivo o causa del rechazo.

- Un partner o un colaborador pueden solicitar crear, modificar o eliminar un cliente.



Cuando la petición llegue al administrador tendrá que aprobarla o rechazarla. Si la aprueba se enviará una notificación al partner o colaborador. A su vez cuando el partner o colaborador lea la notificación del canal interno se enviará una confirmación de la lectura.

- Un administrador podrá emitir facturas a los clientes o partners.



Cuando se emite una factura a un cliente, podrá aprobarla o rechazarla. Si la aprueba se emitirá un cobro bancario (proceso que realizarán manualmente los administradores) y automáticamente se realizará la ampliación correspondiente del tiempo de caducidad de la cuenta. En caso de rechazo se notificará a administración para que quede constancia del rechazo y si procede se pongan en contacto con el cliente.

En el caso de los partners se emitirá una factura por una cuenta o conjunto de cuentas dependiendo la coincidencia o no de la caducidad. El proceso será el mismo, si se acepta se procederá con la ampliación automática del tiempo, sino se notificará a administración para ponerse en contacto.

Los clientes también tendrán acceso a la parte de mensajes internos pero no al chat. La línea de chat es una línea de comunicación entre colaborador, partners y administración.

Mecanismos de Alerta.

Cuando un administrador, cliente, colaborador o partner no conteste o interactue con alguna de las comunicaciones anteriores y pasen 24 horas desde la notificación, el sistema enviará una alerta a través de email(en un futuro también se realizaría a través de sms) que indique a cada uno de estos usuarios que si no realiza ninguna acción en el timeline se realizará una por defecto. Cada caso dependerá de que acción por defecto se ejecutará:

- Un partner o colaborador puede solicitar la activación/desactivación de una cuenta de un cliente.

Alerta a administrador: Un colaborador o partner te ha solicitado hace más de 24 horas que actives una cuenta de cliente. Si no realizas ninguna acción en las próximas 24 horas el sistema activará automáticamente la cuenta.

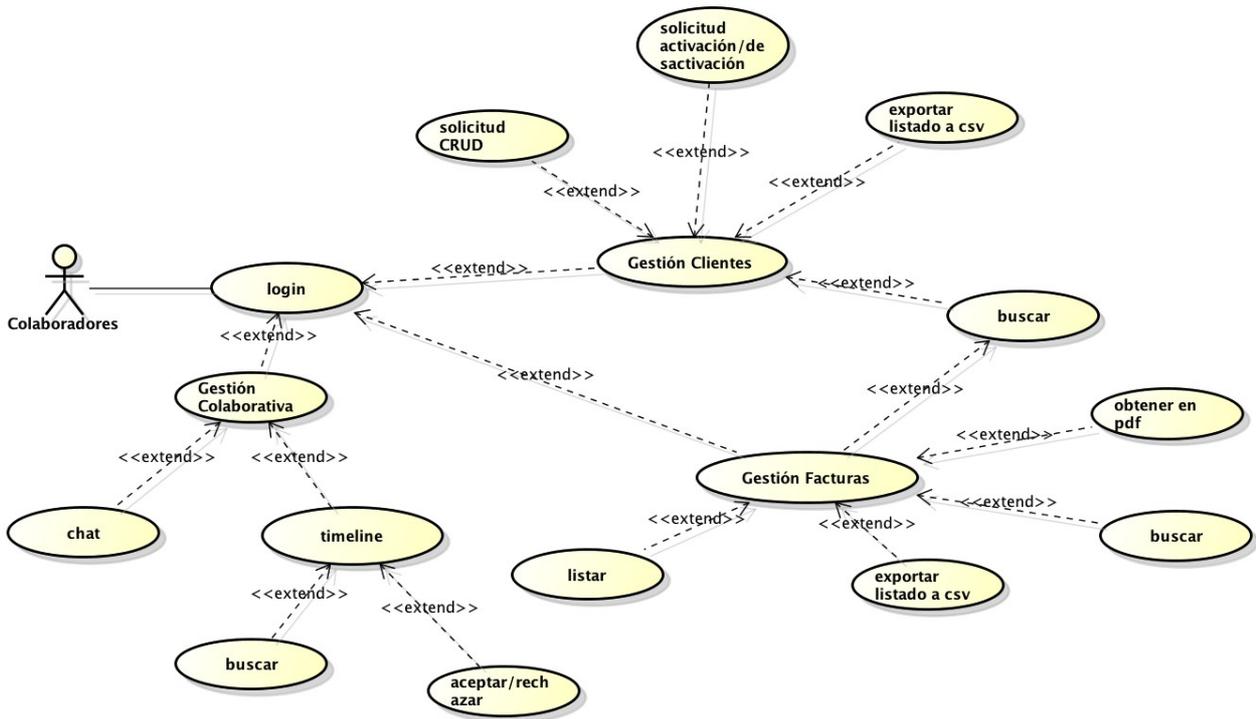
- Un partner o un colaborador pueden solicitar crear, modificar o eliminar un cliente.

Alerta a administrador: Un colaborador o partner te ha solicitado hace más de 24 horas que quiere modificar/eliminar/crear datos de un cliente. Si no realizas ninguna acción en las próximas 24 horas el sistema procederá a actualizar automáticamente la cuenta.

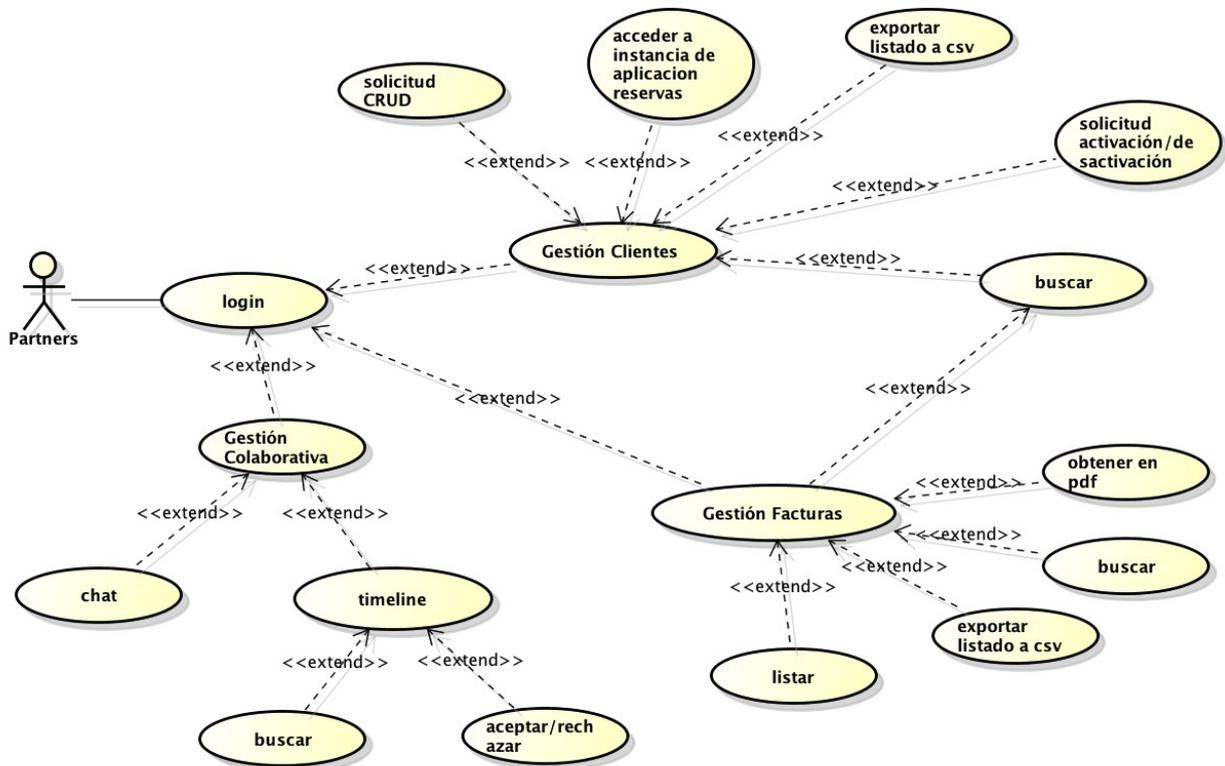
- Un administrador podrá emitir facturas a los clientes o partners.

Alerta a cliente o partner: El administrador del sistema ha emitido una factura a tu cuenta hace más de 24 horas. Si en las próximas 24 horas no realizas ninguna acción sobre ella el sistema aceptará automáticamente la factura.

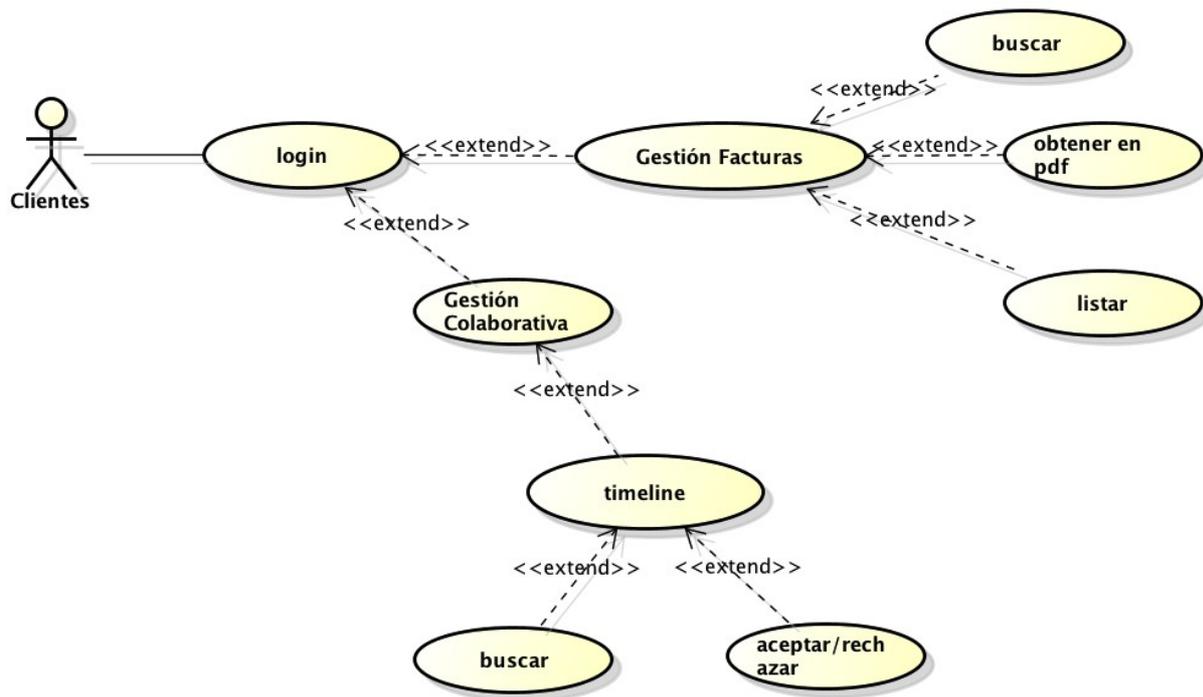
Casos de uso para el actor Colaborador



Casos de uso para el actor Partner



Casos de uso para el actor Cliente



5.1.2.2 Casos de Uso Detallados.

En esta sección se explicará de forma detallada cada uno de los casos de uso mostrados en los gráficos anteriores. Para la elaboración de los casos de uso se ha utilizado un formato de actores, descripción, precondition, postcondition, flujo normal, flujo alternativo.

Caso de uso Login.

Actores: Administrador, colaborador, partner, cliente.

Descripción: Un usuario independientemente del tipo que sea, tendrá que validarse para poder acceder al sistema y realizar acciones.

Precondición: El usuario no está validado.

Flujo principal:

1. Usuario accede a la pantalla de validación introduce su email y su password.
2. Accede al sistema.

Flujo Alternativo:

1. Usuario accede a la pantalla de validación introduce su email y password.
2. El sistema muestra un fallo por usuario incorrecto.

Caso de uso Gestión de Clientes.

Actor: Administrador, colaborador, partner.

Descripción:

Un administrador, un colaborador o un partner, tras validarse correctamente en la aplicación, podrán acceder a la gestión de clientes. Esta gestión consistirá en un listado de los clientes que tiene el sistema (en el caso de administradores). En el caso de colaboradores y partners mostrará un listado de clientes que están asociados a sus cuentas, es decir serán clientes que han sido creados por ellos mismos. El único usuario que podrá ver todos los clientes del sistema será el actor administrador. Desde este listado se mostrarán más acciones que podrán ser realizadas sobre cada uno de los usuarios como son, buscar, crear, modificar, eliminar nuevos clientes (CRUD create, retrieve, update, delete), exportar listados en csv o activar/desactivar una cuenta.

Precondición:

1. El usuario tiene que estar validado correctamente en el sistema.

Postcondición:

No existe.

Flujo Principal:

1. La aplicación accede a la base de datos para obtener un listado de clientes y lo muestra en forma de listado.
2. Además se muestran al usuario acciones que pueda realizar sobre el listado obtenido.

Flujo Alternativo:

No existe.

Caso de uso Acceder a instancia de aplicación de reservas.

Actor: Administrador, partner.

Descripción:

Un administrador o un partner podrán acceder a la instancia de la aplicación de reservas del cliente que haya seleccionado en el listado. El acceso a las instancias de la aplicación de reservas de cada cliente sirve para resolver incidencias o ayudar en la configuración de una cuenta a un cliente. Desde el listado de clientes un administrador o partner podrá acceder a la instancia del cliente haciendo clic en un icono de acceso a instancia.

Precondición:

1. Se tendrá que acceder desde el listado de clientes.

Postcondición:

1. Acceso a la cuenta del cliente sin tener que validarse de nuevo.

Flujo Principal:

1. El usuario visualiza el listado de clientes. En la zona de la derecha podrá ver una serie de acciones que podrá realizar sobre cada uno de los clientes.
2. El usuario hace clic en acceder a instancia.
3. Se abre una nueva ventana en el navegador web con la instancia del acceso a la cuenta del cliente.

Flujo Alternativo:

1. El usuario visualiza el listado de clientes. En la zona de la derecha podrá ver una serie de acciones que podrá realizar sobre cada uno de los clientes.
2. El usuario hace clic en acceder a instancia.
3. Se produce un error de validación cuando se intenta realizar el acceso.

Caso de uso Activar/desactivar cuenta de cliente.

Actor: Administrador

Descripción:

Un administrador podrá activar o desactivar la cuenta de un cliente. Cuando una cuenta de un cliente está desactivada lo que indica es que el cliente no tiene acceso o que no puede validarse para acceder a la herramienta aunque disponga de cuenta. Esto puede permitir a un administrador retener una cuenta de un cliente por ejemplo en el caso de que no pague una cuota. Para realizarlo bastará con hacer clic en un icono indicativo desactivar desde el listado de clientes. No hay que confundir este caso de uso con la “solicitud de activar/desactivar cuenta” que explicaremos más adelante para partners y colaboradores.

Precondición:

1. Se tendrá que acceder desde el listado de clientes.

Postcondición:

No existe.

Flujo Principal:

1. El usuario visualiza el listado de clientes. En la zona de la derecha podrá ver una serie de acciones que podrá realizar sobre cada uno de los clientes.
2. El usuario hace clic en activar/desactivar.
3. Se muestra un mensaje al usuario indicando que la cuenta ha sido desactivada.

Flujo Alternativo:

1. El usuario visualiza el listado de clientes. En la zona de la derecha podrá ver una serie de acciones que podrá realizar sobre cada uno de los clientes.
2. El usuario hace clic en activar/desactivar.
3. Se muestra un mensaje de error indicando que no se ha podido realizar la acción sobre el usuario.

Caso de uso exportar listado a csv.

Actor: Administrador, colaborador, partner.

Descripción:

Tanto administradores, colaboradores como partners tendrán la opción de exportar listados a formato csv. La mayor parte de las ventanas principales de la aplicación son listados los cuales pueden ser interesantes exportar un formato compatible con excel. Una vez accedido al listado aparecerá un botón que permitirá exportar cada uno de los listados en este formato.

Precondición:

1. Se tendrá que acceder desde un listado.

Postcondición:

No existe.

Flujo Principal:

1. El usuario visualiza un listado. En la zona de la derecha podrá ver una serie de acciones que podrá realizar sobre cada uno de los elementos que se mostrará.
2. El usuario hace clic en exportar a csv.
3. Se muestra un mensaje con la descarga del archivo.

Flujo Alternativo:

1. El usuario visualiza un listado. En la zona de la derecha podrá ver una serie de acciones que podrá realizar sobre cada uno de los elementos que se mostrará.
2. El usuario hace clic en exportar a csv.
3. Se muestra un mensaje de error indicando al usuario que no se ha podido generar el archivo.

Caso de uso buscar.

Actor: Administrador, colaborador, partner, clientes.

Descripción:

Tanto administradores, colaboradores como partners y clientes tendrán la opción de realizar búsquedas sencillas en los diferentes listados que muestra la aplicación. Estos buscadores dispondrán de un campo para introducir el texto que se quiere buscar y un desplegable para indicar porque campo se debería realizar la búsqueda. Los campos por los que se podrá realizar la búsqueda dependerá de la situación o ventana en la que se encuentren pero siempre serán los mismos campos que los que se muestren en los listados.

Precondición:

1. Se tendrá que acceder desde un listado.

Postcondición:

No existe.

Flujo Principal:

1. El usuario visualiza el listado. En la zona de la derecha podrá ver una serie de acciones que podrá realizar sobre cada uno de los elementos.
2. El usuario escribe el texto que quiere buscar en la casilla correspondiente.
3. El usuario selecciona el campo por el que quiere realizar la búsqueda.
4. Se muestran los resultados referentes al texto introducido por el usuario.

Flujo Alternativo:

1. El usuario visualiza el listado. En la zona de la derecha podrá ver una serie de acciones que podrá realizar sobre cada uno de los elementos.
2. El usuario escribe el texto que quiere buscar en la casilla correspondiente.
3. El usuario selecciona el campo por el que quiere realizar la búsqueda.
4. Se muestra no hay resultados en el caso de que no los haya.

Caso de uso CRUD Cliente(CREATE, RETRIEVE, UPDATE AND DELETE).

Actor: Administrador.

Descripción:

Desde los listados se podrá realizar acciones para añadir, modificar, eliminar o ver. En el caso de los clientes, se podrá añadir un cliente haciendo clic en el botón de clientes. Para completar el alta se solicitarán los siguientes campos nombre empresa, nombre, apellidos, cif, email, móvil, teléfono, país, región, ciudad, código postal, dirección. También se podrán indicar los detalles de la cuenta, número de agendas disponibles, sms, producto, días de prueba y fecha de fin de cuenta. La ventana de modificación o update también tendrá los mismos campos. Finalmente desde el listado de clientes en la zona de acciones se podrá eliminar un cliente.

Precondición:

1. Se tendrá que acceder desde un listado.

Postcondición:

No existe.

Flujo Principal:

1. El usuario visualiza el listado. En la zona de la izquierda podrá ver una acción para añadir un nuevo cliente.
2. Al hacer clic se visualizarán todos los campos para dar de alta un cliente. Se marcarán algunos campos como obligatorios; Nombre de empresa, email, password, país, región, ciudad, teléfono móvil, número de agendas, tipo de producto y fecha de fin de cuenta.

3. El usuario hace clic en guardar y se muestra el mensaje de guardado correctamente.

Flujo Alternativo:

1. El usuario visualiza el listado. En la zona de la izquierda podrá ver una acción para añadir un nuevo cliente.
2. Al hacer clic se visualizarán todos los campos para dar de alta un cliente.
3. El usuario hace clic en guardar pero tiene campos sin completar, se mostrará un error para que el usuario complete los campos necesarios.

Caso de uso Gestión de Facturas.

Actor: Administrador.

Descripción:

Un administrador tras validarse correctamente en la aplicación, podrán acceder a la gestión de facturas. El administrador observará un listado de facturas al cual se podrá acceder de dos formas, la primera pinchando en la opción del menú general de gestión de facturas, donde las facturas saldrán ordenadas ordenadas por fecha de emisión, la segunda a partir del listado de clientes a través de un acceso que filtre las facturas de ese cliente.

En ambos casos se establecerá un código de colores para una identificación más sencilla de las facturas que están cobradas y facturas que están pendientes de pago. Además también se podrán crear y emitir nuevas facturas a los clientes, actualizar facturas y eliminar facturas. Las facturas deberán contener los datos del cliente(normalmente una empresa) nombre, cif, dirección, ciudad, país, localidad. Las facturas podrán ser emitidas con IVA o sin IVA ya que en ocasiones los clientes suelen ser de países distintos a España o regiones a las que no se cobra iva. También deberán disponer la posibilidad de realizar un descuento.

Además se podrán crear nuevas facturas, actualizar los datos de las facturas y eliminar facturas en un momento determinado.

Precondición:

1. El usuario tiene que estar validado correctamente en el sistema.

Postcondición:

No existe.

Flujo Principal:

1. El usuario hace clic en gestión de facturas.
2. La aplicación consulta la base de datos y obtiene todas las facturas.
3. Además se muestran al usuario acciones que pueda realizar sobre el listado obtenido como crear facturas, modificar factura o eliminar factura.

Flujo Alternativo:

1. El usuario elige un cliente.
2. El usuario pincha en ver facturas del cliente.
3. Se muestra un listado de las facturas asociadas al cliente y sobre el que puede realizar una serie de acciones como eliminar, modificar o crear facturas.

Caso de uso CRUD Facturas(CREATE, RETRIEVE, UPDATE AND DELETE).

Actor: Administrador.

Descripción:

Desde los listados se podrán realizar acciones para añadir, modificar, eliminar o ver facturas. En el caso de las facturas un usuario administrador podrá crear una factura y emitirla al cliente. Los datos de la factura estarán compuestos por email, nombre de empresa, nombre, cif, pais, region, ciudad, código postal y dirección. Además se tendrá que seleccionar el tipo de producto para el que se está haciendo la factura, el concepto, el descuento, el número de factura, el precio y el iva. Desde el apartado de modificación también se podrán cambiar todos estos valores.

Una vez introducidos los datos el usuario administrador podrá enviar la factura por email y pasarla a modo colaborativo con el cliente. En el caso colaborativo un cliente tendrá que aceptar la factura para que se active su cuenta, en caso contrario su cuenta permanecerá temporalmente desactivada.

Precondición:

1. Se tendrá que acceder desde un listado y estar haber hecho login en la aplicación.

Postcondición:

No existe.

Flujo Principal:

1. El usuario visualiza el listado. En la zona de la izquierda podrá ver una acción para añadir una nueva factura.
2. Al hacer clic se visualizarán todos los campos para dar de alta una factura. Se marcarán algunos campos como obligatorios; todos los campos serán obligatorios excepto cif/nif descuento e iva.
3. El usuario hace clic en enviar factura por email y colaborativo.
4. Se enviará un email al cliente indicando que tiene una factura disponible en su zona de facturas para aceptar o rechazar.

Flujo Alternativo 1

1. El usuario visualiza el listado. En la zona de la izquierda podrá ver una acción para añadir una nueva factura.
2. Al hacer clic se visualizarán todos los campos para dar de alta una factura. Se marcarán algunos campos como obligatorios; todos los campos serán obligatorios excepto cif/nif

descuento e iva.

3. Usuario hace clic en guardar y se guarda la factura.

Flujo Alternativo 2

1. El usuario visualiza el listado. Selecciona una factura.
2. Hace clic en eliminar.
3. Al hacer clic se pregunta al usuario si desea eliminarla.
4. Se hace clic en aceptar y se elimina la factura.

Caso de uso obtener pdf.

Actor: Administrador, partner, colaborador, cliente

Descripción:

Desde el listado de facturas se pueden obtener copias en pdf de las facturas emitidas. Un administrador podrá obtener copia del cliente que quiera, un colaborador o partner sólo podrán obtener facturas de sus clientes, un cliente sólo podrá obtener facturas propias. Para realizarlo simplemente aparecerá un icono en el listado de acciones de las facturas donde podrán hacer clic para obtener la factura seleccionada.

Precondición:

1. Se tendrá que acceder desde un listado y haber hecho login en la aplicación.

Postcondición:

2. Se necesita una aplicación externa que permita abrir documentos en formato pdf.

Flujo Principal

1. El usuario visualiza el listado y visualmente escoge una factura.
2. El usuario hace clic en la opción de obtener factura en formato pdf.
3. Se abrirá una ventana para proceder con la descarga de la factura.

Flujo Alternativo

1. El usuario visualiza el listado y visualmente escoge una factura.
2. El usuario hace clic en la opción de obtener factura en formato pdf.
3. Al hacer clic se provoca un error y se indica al usuario que no se ha podido obtener la factura en formato pdf.

Caso de uso Gestión de Colaboradores/Partners.

Actor: Administrador.

Descripción:

Un administrador podrá acceder al listado de colaboradores y partners para gestionarlos. En este listado se podrá buscar, crear, modificar o eliminar partners. El listado podrá ser ordenado por los diferentes campos por los que se mostrará.

Precondición:

1. El usuario tiene que estar validado correctamente en el sistema.

Postcondición:

No existe.

Flujo Principal:

1. La aplicación accede a la base de datos para obtener un listado de partners y colaboradores y lo muestra en forma de listado.
2. Además se muestran al usuario acciones que pueda realizar sobre el listado obtenido.

Flujo Alternativo:

No existe.

Caso de uso CRUD Colaboradores/Partners(CREATE, RETRIEVE, UPDATE AND DELETE).

Actor: Administrador.

Descripción:

Desde este listado se podrán realizar acciones para añadir, modificar, eliminar o ver partners/colaboradores. Por defecto ver coincidirá con el listado estándar. La opción añadir se utilizará para crear nuevos colaboradores o partners, además de poder modificarlos o eliminarlos. Cuando se elimina un partner o colaborador que tiene asociados clientes, estos pasarán a estar asociados a clientes de administración.

Cuando se cree o modifique un partner o colaborador se podrán introducir los siguientes campos: Nombre de empresa, nombre de contacto, cif/nif, pais, tipo de facturación, región, ciudad, código postal, dirección, email de acceso, password, estado activado/desactivado, tipo.

Además se indicará si el partner o colaborador puede acceder a las cuentas para ofrecer soporte. Aunque inicialmente los colaboradores no tienen que dar el soporte es posible que haya ocasiones en las que sea necesario, pero no será lo habitual.

Precondición:

1. Se tendrá que acceder desde un listado.

Postcondición:

No existe.

Flujo Principal

1. El usuario visualiza el listado. En la zona de la izquierda podrá ver una acción para añadir un nuevo colaborador o partner.
2. Al hacer clic se visualizarán todos los campos para dar de alta el partner o colaborador. Los campos del colaborador serán todos obligatorios a excepción de código postal, dirección y el acceso a las cuentas de clientes.
3. El usuario hace clic guardar y crea la cuenta de colaborador o partner.

Flujo Alternativo

1. El usuario visualiza el listado. En la zona de la izquierda podrá ver una acción para añadir un nuevo colaborador o partner.
2. Al hacer clic se visualizarán todos los campos para dar de alta el partner o colaborador. Los campos del colaborador serán todos obligatorios a excepción de código postal, dirección y el acceso a las cuentas de clientes.
3. Al hacer clic en guardar se produce un error y se muestra al usuario.

Caso de uso solicitud de CRUD clientes(CREATE, RETRIEVE, UPDATE AND DELETE).

Actor: Colaborador, Partners.

Descripción:

Un colaborador o un partner podrá en cualquier momento introducir los datos de un cliente y solicitar al administrador su aceptación o rechazo. Todas estas acciones se verán reflejadas en el timeline de colaboración. Además podrá solicitar eliminar un cliente, que haya creado, o una modificación. De esta forma siempre las acciones que realice sobre los clientes tendrás que estar supervisadas por un administrador.

Precondición:

1. Se tendrá que acceder desde un listado de clientes.

Postcondición:

No existe.

Flujo Principal

1. El partner/colaborador visualiza el listado. En la zona de la izquierda podrá ver una acción para añadir un nuevo cliente.
2. Al hacer clic se visualizarán todos los campos para dar de alta el cliente, introducirá los campos (los campos obligatorios serán los mismo que para el caso de un administrador) y solicitará su aprobación por parte del administrador.
3. El administrador recibe la notificación y acepta la creación del cliente.
4. El sistema almacena la información y notifica al colaborador o partner que el alta se ha completado con éxito.

Flujo Alternativo 1.

1. El partner/colaborador visualiza el listado. En la zona de la izquierda podrá ver una acción para añadir un nuevo cliente.
2. Al hacer clic se visualizarán todos los campos para dar de alta el cliente, introducirá los campos (los campos obligatorios serán los mismo que para el caso de un administrador) y solicitará su aprobación por parte del administrador.
3. El administrador recibe la notificación y rechaza la creación del cliente y especifica el motivo.
4. El sistema notifica al solicitante que se ha rechazado su petición y observa el motivo (timeline).
5. El colaborador vuelve acceder a modificar el cliente, realiza el cambio solicitado por el administrador y vuelve a emitir la petición.
6. El administrador vuelve a recibir la notificación y acepta el cambio.

Flujo Alternativo 2.

1. El partner/colaborador visualiza el listado y escoge un cliente.
2. El partner/colaborador realiza la solicitud de eliminar el cliente.
3. El sistema envía la petición al administrador que comprueba si realmente se tiene que completar esa eliminación.
4. El administrador acepta la eliminación y se procede a dar de baja el cliente.

Caso de uso solicitud de activar/desactivar clientes.

Actor: Resellers, Partners

Descripción:

Un colaborador o un partner podrá en cualquier momento solicitar la activación o desactivación de un cliente a los administradores. Este caso puede ser debido a un impago o incumplimiento de algún tipo de condición.

Precondición:

1. Se tendrá que acceder desde un listado de clientes.

Postcondición:

No existe.

Flujo Principal

1. El partner/colaborador visualiza el listado. En la zona de la izquierda podrá ver una acción para desactivar el cliente.
2. Al hacer clic en la activación/desactivación se enviará una notificación a través de email y timeline al administrador.
3. El administrador recibe la notificación y acepta la activación del cliente.
4. El sistema almacena la información y notifica al colaborador o partner que el activación se ha completado con éxito.

Flujo Alternativo 1.

1. El partner/colaborador visualiza el listado. En la zona de la izquierda podrá ver una acción para activar/desactivar un nuevo cliente.
2. Al hacer clic en la activación/desactivación se enviará una notificación a través de email y timeline al administrador.
3. El administrador recibe la notificación y rechaza la activación del cliente, introduciendo el motivo.
4. El sistema almacena la información y notifica al colaborador o partner que la activación no se ha completado por el motivo introducido por el administrador.

Caso de uso gestión colaborativa.

Actor: Administrador, clientes, colaboradores, partners.

Descripción:

Administradores, clientes, resellers y partners pueden interactuar de forma colaborativa para realizar diversas acciones sobre la información del sistema. Estas acciones se verán reflejadas en un timeline del sistema donde podrán interactuar acciones de aceptación, rechazo o informativas al resto de usuarios.

Precondición:

1. Se tendrá que acceder una vez el usuario haya hecho login.

Postcondición:

No existe.

Flujo Principal

1. Usuario hace clic en gestión colaborativa.
2. El sistema obtiene los mensajes que tiene que mostrar al usuario.

Caso de uso chat.

Actor: Administrador, clientes, colaboradores, partners.

Descripción:

Tanto clientes, colaboradores, partners y administradores podrán colaborar a través de un chat interno que dispondrá la aplicación.

Precondición:

1. Se tendrá que acceder una vez esté validado el usuario.

Postcondición:

No existe.

Flujo Principal

1. El usuario hace clic en la opción de acceder a chat.
2. El usuario visualiza los usuarios que están conectados al chat.
3. El usuario selecciona el destinatario al que quiere enviar los mensajes.
4. El usuario escribe el mensaje que desea enviar.

Caso de uso timeline.

Actor: Clientes, administradores, partners y colaboradores.

Descripción:

Cada vez que se realiza una acción colaborativa en el sistema quedará reflejada en el timeline de cada usuario correspondiente. Además se podrá interactuar sobre cada una de las acciones colaborativas. Como acciones interactivas se pueden detectar, solicitud de creación/modificación/eliminación de un cliente, solicitud de activación/desactivación de un cliente, creación de una factura a un cliente. Cada una de estas acciones podrá ser aceptada o rechazada a través de un enlace que aparecerá en el listado del timeline.

Precondición:

1. Se tendrá que acceder una vez esté validado el usuario.

Postcondición:

No existe.

Flujo Principal

1. El usuario visualiza el timeline de acciones.
2. El usuario acepta o aprueba una acción.
3. El sistema notifica al usuario solicitante y lo refleja en su timeline.

5.1.3 Diseño.

5.1.3.1 Diseño de Interfaces.

ADMINISTRADORES

Pantalla de Login

A Web Page

Backend Access

Usuario:

Password:

Pantalla de Gestión de Clientes

A Web Page

[Clientes](#) [Facturas](#) [Colaboradores](#) [Gestión Colaborativa](#) Bienvenido Usuari@ [Cerrar Sesión](#)

[Clientes](#) > ...

Cientes

Id	Nombre	Email	Telefono	Fecha Alta	Pais	SMS	Producto Contratado	Reservas Último Mes	Fecha de Renovación	Estado de Renovación	Acciones
1	Fisio Juan	jhont@mensa.com	99999999	27-11-2012	España	33	Standard	22	27-11-2013	Pago por transferencia	Acceder facturas Off Mod del
2	Fisio Juan	jhont@mensa.com	99999999	27-11-2012	España	33	Standard	22	27-11-2013	Pagada con tarjeta	Acceder facturas Off Mod del
3	Fisio Juan	jhont@mensa.com	99999999	27-11-2012	España	33	Standard	22	27-11-2013	No realizada	Acceder facturas Off Mod del

Pantalla de creación de un nuevo cliente

A Web Page

Clientes Facturas Colaboradores Gestión Colaborativa Bienvido Usuari@ [Cerrar Sesión](#)

[Clientes](#) > añadir

Nuevo Cliente

Nombre Empresa: Teléfono:

Nombre: País:

Apellidos: Región:

CIF/NIF: Ciudad:

Email: Código Postal:

Móvil: Dirección:

Detalles de cuenta

Nº de Agendas: Producto:

SMS: Trial Days:

Estado: ON Fecha de Alta: 27-11-2012 13:41 Fecha de fin:

Al guardar enviar email con datos de activación y password

Pantalla de actualización de un cliente

A Web Page

Clientes Facturas Colaboradores Gestión Colaborativa Bienvido Usuari@ [Cerrar Sesión](#)

[Clientes](#) > añadir

Nuevo Cliente

Nombre Empresa: Teléfono:

Nombre: País:

Apellidos: Región:

CIF/NIF: Ciudad:

Email: Código Postal:

Móvil: Dirección:

Detalles de cuenta

Nº de Agendas: Producto:

SMS: Trial Days:

Estado: ON Fecha de Alta: 27-11-2012 13:41 Fecha de fin:

Al guardar enviar email con datos de activación y password

Pantalla de gestión de facturas

Facturas > ...

Facturas

Añadir Exportar a csv

Q search id Buscar

Id	Nombre Empresa	Email	Pais	Concepto	Fecha Emisión	Forma de Pago	Estado	Producto Contratado	Nombre Empresa
1	Fisio Juan	jhont@mensa.com	España	Standard x 12 meses	02-02-2012	Paypal	Pagada	Standard	pdf Mod del
2	Pelu Pepi	pelupepi@pp.com	España	Standard x 12 meses	01-01-2012	Paypal	Pagada	Standard	pdf Mod del
3	Spa Solar	info@spasolar.com	España	Standard x 12 meses	25-12-2011	Transferencia	No pagada	Standard	pdf Mod del

Pantalla de nueva factura

Facturas > Nueva Factura > ...

Nueva Factura

Datos del Cliente

Email: Some text Region: ComboBox

Nombre Empresa: Some text Ciudad: ComboBox

Nombre: Some text Código Postal: Some text

Pais: ComboBox Dirección: Some text

CIF/NIF: Some text

Datos de Factura

Producto: Standard, MultiUser N° Factura: Some text

Concepto: Concepto de la factura Precio: Some text

Dto: Some text Iva: Some text

Caducidad: Concepto de la factura Total: xxxxxxx

Al guardar enviar factura por email y colaborativo Guardar Cancelar

Pantalla de actualizar facturas

A Web Page

Clientes Facturas Colaboradores Gestión Colaborativa Bienvenido Usuari@ [Cerrar Sesión](#)

Facturas > Nueva Factura > ...

Nueva Factura

Datos del Cliente

Email: Region:

Nombre Empresa: Ciudad:

Nombre: Código Postal:

País: Dirección:

CIF/NIF:

Datos de Factura

Producto: N° Factura:

Concepto: Precio:

Dto: Iva:

Caducidad: Total: xxxxxxx

Al guardar enviar factura por email y colaborativo

Pantalla de gestión de Colaboradores/Partners

A Web Page

Clientes Facturas Colaboradores Gestión Colaborativa Bienvenido Usuari@ [Cerrar Sesión](#)

Colaboradores > ...

Colaboradores

Id	Nombre Empresa	Nombre Contacto	Email Contacto	Teléfono	Ciudad	País	CIF	Tipo	Cientes	Facturación	Nombre Empresa Email Contacto ...
1	ComerciaSoft	Juan Benito	jbenito@comerciasoft.com	65543321	Valencia	España	23423432G	Partner	28	15000	Modificar Eliminar
2	VendeSoft	Jose Vende	jvende@vendesoft.com	623434444	Valencia	España	23423222G	Partner	5	2000	Modificar Eliminar
3	Manolo Autónomo	Manolo Autónomo	jbenito@comerciasoft.com	65543321	Valencia	España	23423432G	Colaborador	28	15000	Modificar Eliminar

Pantalla de nuevo colaborador/partner

Pantalla de modificar colaborador/partner

Pantalla gestión colaborativa timeline

The screenshot shows a web browser window titled "A Web Page" with a navigation bar containing buttons for "Clientes", "Facturas", "Colaboradores", and "Gestión Colaborativa (5 mensajes nuevos)". A user is logged in as "Bienvenido Usuari@" with a "Cerrar Sesión" link. The main content area is titled "Gestión Colaborativa" and includes an "Acceder a chat" button. A search bar with a "Buscar" button is present. Below is a table of messages:

Mensaje	Fecha	
El cliente ha aceptado la Factura 01-03 del cliente "Fisioterapia Juan" con email juan@fisioterapia.com	03-04-2012 16:30	
El administrador ha emitido una Factura con código 01-03 al cliente "Fisioterapia Juan" con email juan@fisioterapia.com	03-04-2012 15:30	
El administrador ha emitido una Factura al cliente "Medicina Pepe" con email pepe@medicinapepe.com	02-04-2012 15:30	
El colaborador "ComerciaSoft" ha solicitado la aprobación del alta de un nuevo cliente cliente@cliente.com	01-04-2012 12:30	aceptar rechazar
El colaborador "ComerciaSoft" quiere eliminar la cuenta del cliente cliente2@cliente2.com	01-04-2012 11:30	aceptar rechazar

Pantalla de gestión colaborativa chat

The screenshot shows a web browser window titled "A Web Page" with the same navigation bar as the previous page. The main content area is titled "Gestión Colaborativa" and includes an "Accede a timeline mensajes" button. Below this is a section titled "Selecciona un usuario con el que quieras hablar:" with a list of users: "Usuario A", "Usuario B", "Usuario C", "Usuario D", and "Usuario E". To the right is a chat area with the following text:

Usuario A:
 Como puedo
 Tú:
 No puedes realizar esa acción

COLABORADORES

Pantalla de gestión de clientes

A Web Page

Clientes Facturas Gestión Colaborativa Bienvenido Usuari@ [Cerrar Sesión](#)

Clientes > ...

Clientes

Añadir Exportar a csv Ver todos(listar) search ComboBox Buscar

Id	Nombre	Email	Telefono	Fecha Alta	Pais	SMS	Producto Contratado	Reservas Último Mes	Fecha de Renovación	Estado de Renovación	Acciones
1	Fisio Juan	jhont@mensa.com	99999999	27-11-2012	España	33	Standard	22	27-11-2013	Pago por transferencia	facturas Off Mod del
2	Fisio Juan	jhont@mensa.com	99999999	27-11-2012	España	33	Standard	22	27-11-2013	Pagada con tarjeta	facturas Off Mod del
3	Fisio Juan	jhont@mensa.com	99999999	27-11-2012	España	33	Standard	22	27-11-2013	No realizada	facturas Off Mod del

Pantalla de solicitar nuevo cliente

A Web Page

Clientes Facturas Gestión Colaborativa Bienvenido Usuari@ [Cerrar Sesión](#)

Clientes > añadir

Nuevo Cliente

Nombre Empresa: Teléfono:

Nombre: País:

Apellidos: Región:

CIF/NIF: Ciudad:

Email: Código Postal:

Móvil: Dirección:

Detalles de cuenta

Nº de Agendas: Producto:

SMS: Trial Days:

Estado: ON Fecha de Alta: 27-11-2012 13:41 Fecha de fin:

Al guardar enviar email con datos de activación y password

Pantalla de Actualizar cliente

A Web Page

Clientes Facturas Gestión Colaborativa Bienvenido Usuari@ Cerrar Sesión

Cientes > modificar

Modificar Cliente

Nombre Empresa: Teléfono:

Nombre: País:

Apellidos: Región:

CIF/NIF: Ciudad:

Email: Código Postal:

Móvil: Dirección:

Detalles de cuenta

Nº de Agendas: Producto:

SMS: Trial Days:

Estado: ON Fecha de Alta: 27-11-2012 13:41 Fecha de fin:

Al guardar enviar email con datos de activación y password

Pantalla de Chat con usuarios

A Web Page

Clientes Facturas Gestión Colaborativa (5 mensajes nuevos) Bienvenido Usuari@ Cerrar Sesión

Gestión Colaborativa > ...

Gestión Colaborativa

Accede a timeline mensajes

Selecciona un usuario con el que quieras hablar:

Usuario A
 Usuario B
 Usuario C
 Usuario D
 Usuario E

Usuario A:
 Como puedo

Tú:
 No puedes realizar esa acción

Pantalla de timeline

Acceder a chat

Mensaje	Fecha	Acciones
El administrador ha emitido una Factura con código 01-03 al cliente "Fisioterapia Juan" con email juan@fisioterapia.com	03-04-2012 15:30	
El administrador ha emitido una Factura al cliente "Medicina Pepe" con email pepe@medicinapepe.com	02-04-2012 15:30	
Has solicitado la aprobación del alta de un nuevo cliente cliente@cliente.com	01-04-2012 12:30	aceptar rechazar
Has solicitado la eliminación de la cuenta del cliente cliente2@cliente2.com	01-04-2012 11:30	aceptar rechazar

Pantalla de Gestión de facturas

Añadir Exportar a csv

Ver todos(listar)

Id	Nombre Empresa	Email	País	Concepto	Fecha Emisión	Forma de Pago	Estado	Producto Contratado	Acciones
1	Fisio Juan	jhant@mensa.com	España	Standard x 12 meses	02-02-2012	Paypal	Pagada	Standard	pdf
2	Pelu Pepi	pelupepi@pp.com	España	Standard x 12 meses	01-01-2012	Paypal	Pagada	Standard	pdf
3	Spa Solar	info@spasolar.com	España	Standard x 12 meses	25-12-2011	Transferencia	No pagada	Standard	pdf

PARTNERS

Pantalla de gestión de facturas

Facturas > ...

Facturas

Añadir Exportar a csv Ver todos(listar) search id Buscar

Id	Nombre Empresa	Email	Pais	Concepto	Fecha Emisión	Forma de Pago	Estado	Producto Contratado	Nombre Empresa
1	Fisio Juan	jhant@mensa.com	España	Standard x 12 meses	02-02-2012	Paypal	Pagada	Standard	pdf
2	Pelu Pepi	pelupepi@pp.com	España	Standard x 12 meses	01-01-2012	Paypal	Pagada	Standard	pdf
3	Spa Solar	info@spasolar.com	España	Standard x 12 meses	25-12-2011	Transferencia	No pagada	Standard	pdf

Pantalla de gestión de clientes

Clientes > ...

Clientes

Añadir Exportar a csv Ver todos(listar) search ComboBox Buscar

Id	Nombre	Email	Telefono	Fecha Alta	Pais	SMS	Producto Contratado	Reservas Último Mes	Fecha de Renovación	Estado de Renovación	Acciones
1	Fisio Juan	jhant@mensa.com	99999999	27-11-2012	España	33	Standard	22	27-11-2013	Pago por transferencia	acceder facturas Off Mod del
2	Fisio Juan	jhant@mensa.com	99999999	27-11-2012	España	33	Standard	22	27-11-2013	Pagada con tarjeta	acceder facturas Off Mod del
3	Fisio Juan	jhant@mensa.com	99999999	27-11-2012	España	33	Standard	22	27-11-2013	No realizada	acceder facturas Off Mod del

Pantalla de solicitar nuevo cliente

A Web Page
http://

Clientes
Facturas
Gestión Colaborativa

Bienvenido Usuari@ [Cerrar Sesión](#)

[Clientes](#) > añadir

Nuevo Cliente

Nombre Empresa:

Nombre:

Apellidos:

CIF/NIF:

Email:

Móvil:

Teléfono:

País:

Región:

Ciudad:

Código Postal:

Dirección:

Detalles de cuenta

Nº de Agendas:

SMS:

Estado: ON

Producto:

Trial Days:

Fecha de fin:

Fecha de Alta: 27-11-2012 13:41

Al guardar enviar email con datos de activación y password
 Guardar y solicitar cambios
Cancelar

Pantalla de solicitar actualización de cliente

A Web Page
http://

Clientes
Facturas
Gestión Colaborativa

Bienvenido Usuari@ [Cerrar Sesión](#)

[Clientes](#) > modificar

Modificar Cliente

Nombre Empresa:

Nombre:

Apellidos:

CIF/NIF:

Email:

Móvil:

Teléfono:

País:

Región:

Ciudad:

Código Postal:

Dirección:

Detalles de cuenta

Nº de Agendas:

SMS:

Estado: ON

Producto:

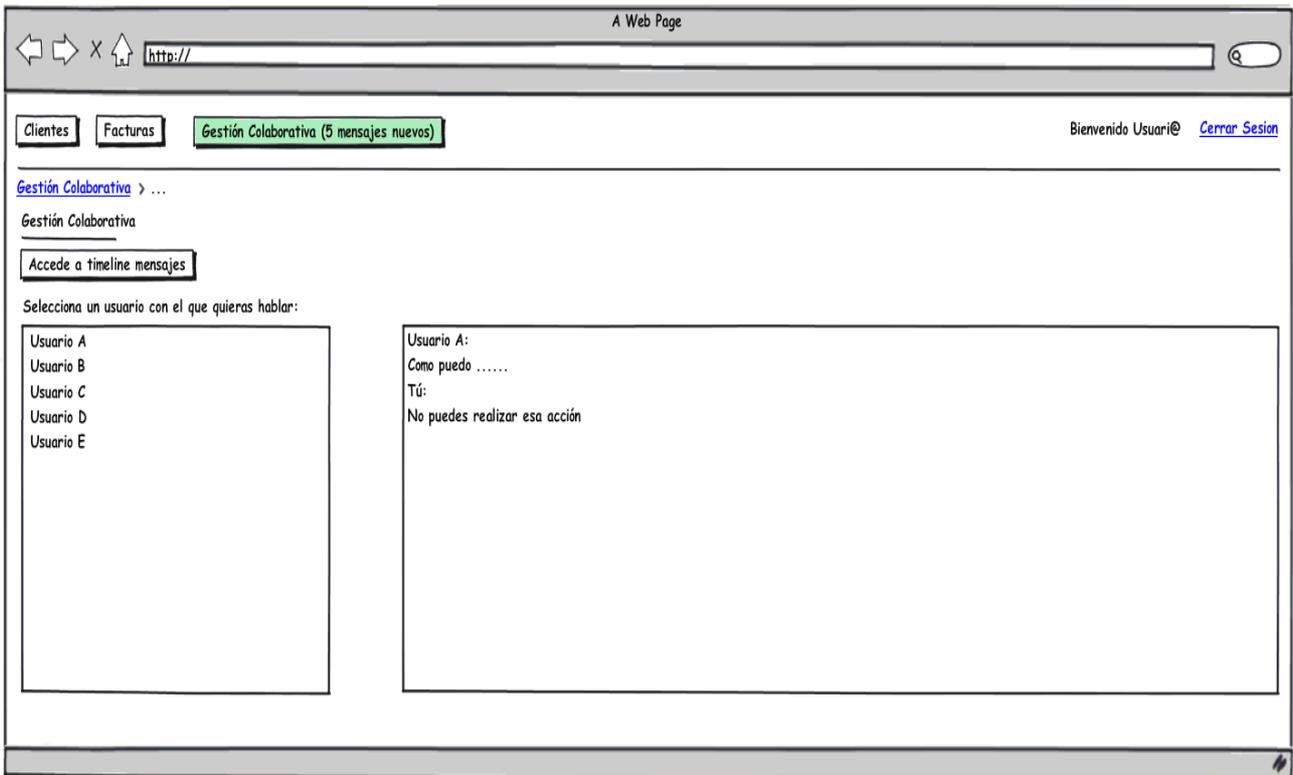
Trial Days:

Fecha de fin:

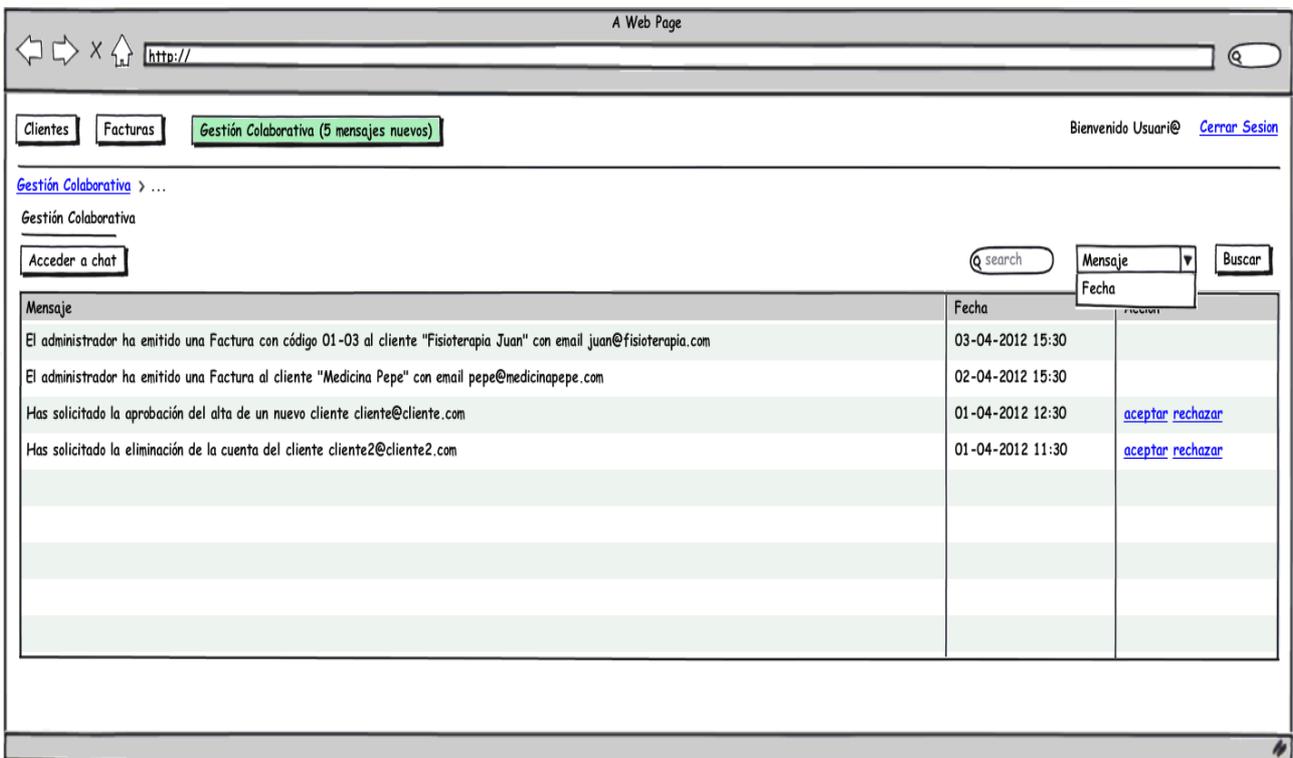
Fecha de Alta: 27-11-2012 13:41

Al guardar enviar email con datos de activación y password
 Guardar y solicitar cambios
Cancelar

Pantalla de chat



Pantalla de timeline (comunicaciones)



CLIENTES

Pantalla de gestión colaborativa

Facturas Gestión Colaborativa (5 mensajes nuevos) Bienenido Usuari@ Cerrar Sesión

Gestión Colaborativa > ...

Gestión Colaborativa

search Mensaje Fecha Buscar

Mensaje	Fecha	Acciones
El administrador ha emitido una Factura con código 01-03 al cliente "Fisioterapia Juan" con email juan@fisioterapia.com	03-04-2012 15:30	aceptar rechazar

Pantalla de descarga de facturas en pdf

Facturas Gestión Colaborativa Bienenido Usuari@ Cerrar Sesión

Facturas > ...

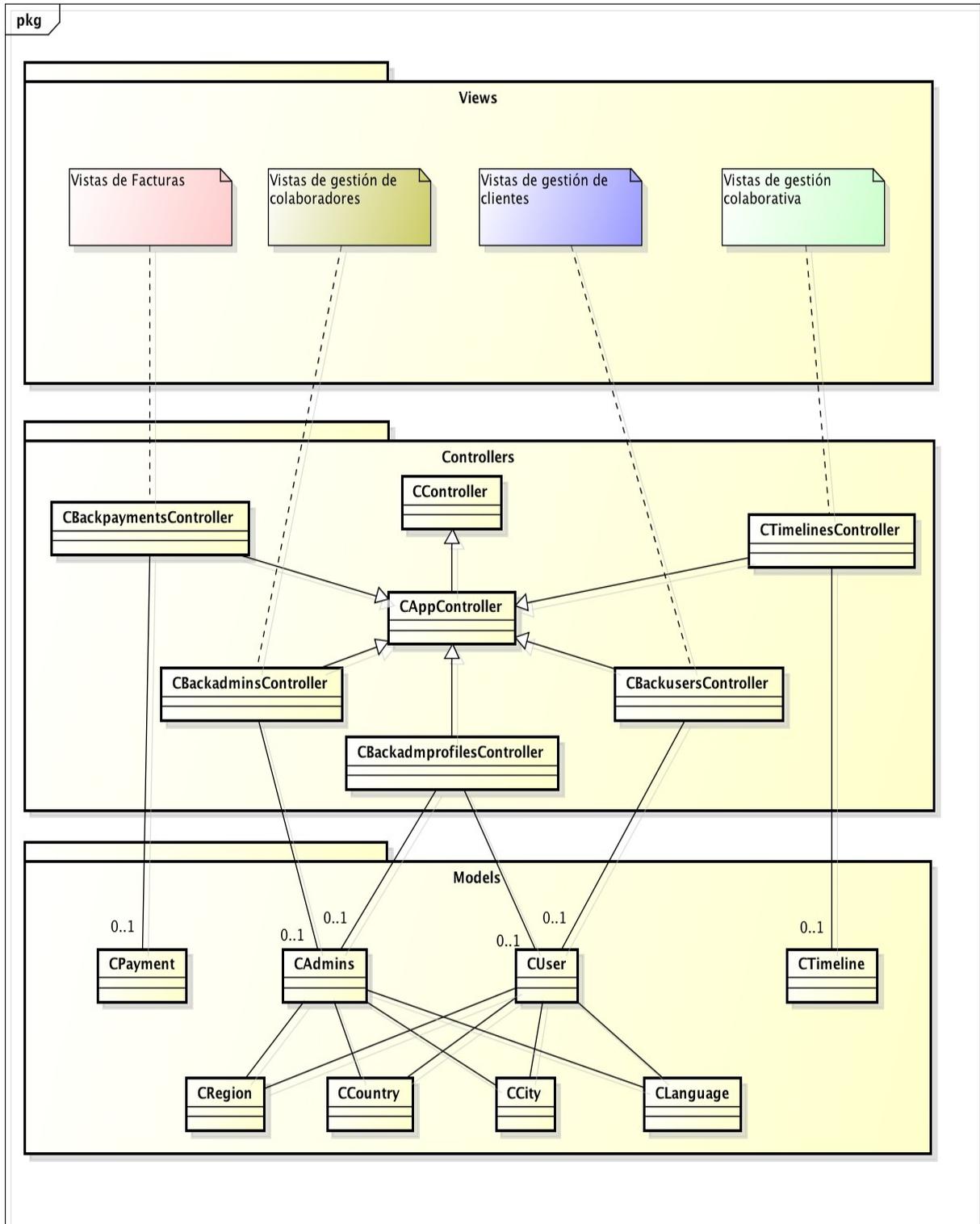
Facturas

Ver todos(listar) search id Nombre Empresa Buscar

Id	Nombre Empresa	Email	Pais	Concepto	Fecha Emisión	Forma de Pago	Estado	Producto Contratado	Nombre Empresa
1	Fisio Juan	jhont@mensa.com	España	Standard x 12 meses	02-02-2012	Paypal	Pagada	Standard	pdf
2	Pelu Pepi	pelupepi@pp.com	España	Standard x 12 meses	01-01-2012	Paypal	Pagada	Standard	pdf
3	Spa Solar	info@spasolar.com	España	Standard x 12 meses	25-12-2011	Transferencia	No pagada	Standard	pdf

5.1.3.2 Diseño de Diagrama de Clases.

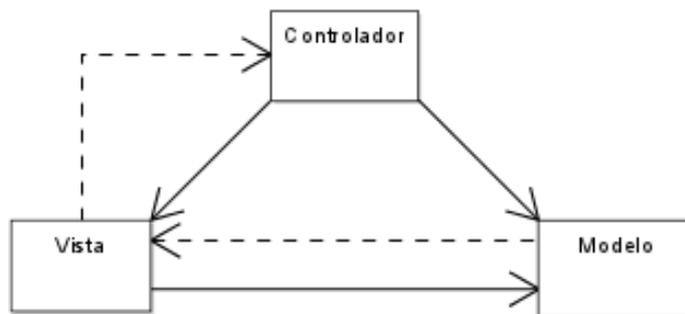
A continuación mostraremos el diseño de diagrama de clases empleado para la sección de backoffice del aplicativo. Como se puede observar se divide en 3 capas basadas en el patrón arquitectónico Modelo-Vista-Controlador.



En cuanto a las vistas podemos decir que se encargarán de todo lo que un usuario puede ver o con lo que interactuar. Cuando un usuario realiza una acción a través de la vista, emite una señal al controlador correspondiente para que desencadene una serie de acciones sobre los modelos.

Estos modelos devolverán a los controladores una serie de información que será mostrada de nuevo al usuario a través de las vistas. Este *workflow* se mantendrá como patrón arquitectónico para toda la aplicación.

Para la implementación del *backoffice* correspondiente al proyecto, se han utilizado 5 controladores. Todos ellos son dependientes de la clase padre CAppController. La clase CAppController es una clase que dota a sus clases hijas de un comportamiento de controlador. A partir de ahí se puede especificar o especializar cada una de forma diferente dependiendo de las acciones para las que nos interese que respondan.



Por ejemplo en el caso CtimelinesController controlará las acciones realizadas en el *timeline* colaborativo de la aplicación, de forma que cuando haya que crear, modificar o notificar será la encargada de recoger la acción y delegarla al modelo que más le interese, en este caso Ctimeline.

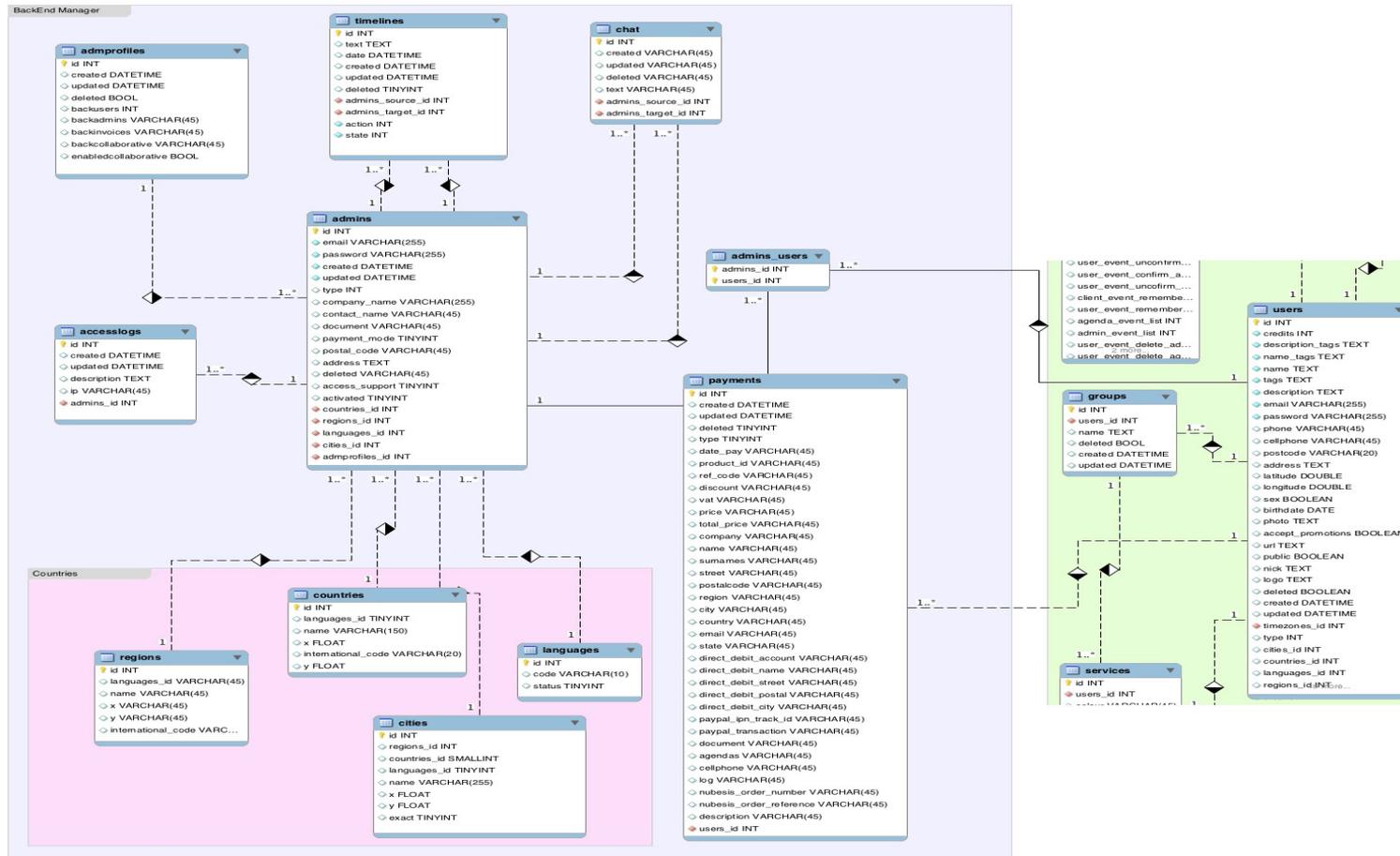
Otro ejemplo podría ser la clase Cbackadmprofiles que se encargará de controlar si un usuario tiene acceso o no para acceder a un área correspondiente.

No olvidemos que en esta aplicación hay distintos perfiles de usuario y cada uno podrá realizar determinadas tareas, por lo tanto habrá que limitar el acceso de cada uno de ellos.

En cuanto a los modelos podemos ver que son usados por la mayoría de controladores y que se encargarán de realizar los accesos a la base de datos o si fuera necesario, a un servicio web para devolver datos en forma de objetos, arrays o cualquier otro tipo de estructura de datos.

5.1.3.3 Diseño del Modelo Físico de la Base de datos.

En esta sección se muestra una imagen del modelo físico entidad relación. En la imagen se muestran las tablas que afectan directamente al módulo que se va a desarrollar para la aplicación. Por motivos de confidencialidad no se mostrará toda la base de datos de la aplicación de reservas online.



Podemos observar que el modelo físico de la base de datos tiene bastante que ver con la capa de modelo de la arquitectura. Normalmente cada una de las clases de modelo hará accesos a la tabla con la que se corresponda. Por ejemplo, podemos ver que existe una tabla admins y una clase de modelo Cadmins.

Entrando un poco más a fondo en el modelo físico vemos que la tabla admins es la más relacionada con el resto de tablas. Se puede observar que para el entorno colaborativo se ha relacionado la tabla admins con timelines y chat. Estas dos tablas tienen dos relaciones ya que normalmente tendrán un usuario origen de la comunicación y un usuario destinatario. Además la tabla admins también estará relacionada con la tabla admprofiles. Esta última es la encargada de gestionar todos los perfiles que pueden acceder a la aplicación además de indicar a que zona pueden acceder en modo lectura, escritura o ambos.

La tabla accesslogs se encarga de gestionar cada uno de los accesos que realizarán los usuarios. Esta tabla es importante ya que de ella podrían depender de auditorías si se produjera algún fallo de seguridad. Nos permitirá conocer cuando y en que momento se han realizado accesos por cada uno de los distintos usuarios.

La tabla payments es la encargada de gestionar todo lo referente a facturación. En ella quedarán reflejados los datos de las facturas, totales, iva, ...

Finalmente comentar la relación la tabla usuarios de la tabla administradores. Hay que tener en cuenta que cuando acceda un colaborador o un partner sólo podrá realizar acciones sobre los usuarios que estén relacionados con él. De este modo un colaborador o partner no podrá realizar acciones sobre clientes de otros usuarios. Esta tabla permite controlar estas acciones.

5.1.4 Implementación.

En esta sección se describirán las funciones, clases o módulos importantes del programa. El software está desarrollado bajo un framework perteneciente a la empresa Z. Este framework está desarrollado bajo una arquitectura modelo-vista-controlador que obliga al desarrollador a seguir unas pautas en la elaboración del código. En concreto hablaremos de 3 secciones importantes en el código. La sección de **control de perfiles**, la sección de **colaboración** y la sección de **gestión de usuarios**.

5.1.4.1 Control de Perfiles.

El control de acceso de los perfiles se encargan de realizarlo básicamente 2 clases. Por un lado la clase CloginsController que es la encargada de realizar las validaciones de cada uno de los distintos usuarios. A continuación podemos ver el método que se encarga de realizar la comprobación de que un existe o no en el sistema.

```
/**
 * This function solves the action /logins/backend. It will be executed
 * when an administrator user will try to log into our system
 */
public function backend() {
    try {

        $oSession = new CAppSession();
        $someParameters = $this->get_someParameters();

        //if you are trying to validate
        if (((isset($someParameters['nameInpPassword'])) &&
            (isset($someParameters['nameInpUser'])))) {

            $sUser = $someParameters['nameInpUser'];
            $sPassword = md5($someParameters['nameInpPassword']);
            $iType = $someParameters["nameSelType"];
            //and if you don't have a session initiated we have to check it
            if ($oSession->__get("typeadmin")==false) {
                //it will ask to table admins
                $oAdmin = $this->getUsedClass("CAdmin");
                $somePartsAdmins = $oAdmin->getSomeAdmins($sUser, $sPassword, $iType);
                if ($somePartsAdmins != null && count($somePartsAdmins) > 0) {
                    if (count($somePartsAdmins) > 0) {
                        //now it will get the profile for Admin
                    }
                }
            }
        }
    }
}
```

```

$soAdmProfile = $this->getUsedClass("CAdmprofile");
$someAdmProfile = $soAdmProfile->getProfileByIdUser($somePartsAdmins[0]['id']);
if (count($someAdmProfile)<=0)
{
    $this->_oLayout->add("sTextMessage",__("This user hasn't a profile",$this->_sLanguage,true));
    $this->_oLayout->add("sTypeMessage",C_MESSAGE_ERROR);
    return false;
}
else
{
    //if get_someData have data values, you will be in the database and we will create the session
    $soSession = new CAppSession();
    $soSession->__set("idadmin",$somePartsAdmins[0]['id']);
    $soSession->__set("loginadmin",$somePartsAdmins[0]['email']);
    $soSession->__set("typeadmin",C_ADMIN_USER);
    $soSession->__set("nameadmin",$somePartsAdmins[0]["contact_name"]);
    $soSession->__set("namecompanyadmin",$somePartsAdmins[0]["company_name"]);
    $soSession->__set("someProfile",$someAdmProfile);
    $soSession->__set("iLanguageId",7); //@todo: change
    $this->_oLayout->add("oSession",$soSession);
    if ($someAdmProfile[0]["id"] == C_PROFILE_CLIENT)
    {
        redirect($this->link("index","timelines"),__("You have been logged in correctly",$this->_sLanguage,true),
C_MESSAGE_OK, true);
    }
    else
    {
        redirect($this->link("index","backusers"),__("You have been logged in correctly",$this->_sLanguage,true),
C_MESSAGE_OK, true);
    }
    return true;
}

}
else {
    //Second we have to set the view that will show the controller
    $this->_oLayout->add("sTextMessage",__("Login or password incorrect",$this->_sLanguage,true));
    $this->_oLayout->add("sTypeMessage",C_MESSAGE_ERROR);
    //$this->_oView->add("sMessage",__("Login or password incorrect",$this->_sLanguage,true));
    return false;
}

```

```

        }
    }
}

//if you arrive here and you have a previous session created
if ($oSession->__get("typeadmin")!=false) {
    redirect($this->link("index","backusers"),__("You have been logged in correctly",$this->_sLanguage,true),
C_MESSAGE_OK, true);
    return true;
}

//if you arrive here and don't have session
$this->_oView->set_sView('backend.php');
$this->_oView->add("sMessage",__("Login or password incorrect",$this->_sLanguage,true));

}
catch(CException $e) {
    throw $e;
}
return false;
}

```

Además esta función se encargará de comprobar si un usuario tiene una sesión iniciada o no para mostrar o no la sección de login cuando se intente acceder sin haber pasado antes por aquí.

Una vez un administrador, colaborador, partner o usuario a accedido al sistema, también deberemos comprobar en cada acción que realice si está accediendo a un controlador y vista que esté permitido. Esto lo realizará la clase CbackadmprofilesController. El framework permite definir en CAppController determinadas llamadas que permitirán evaluar cada petición que se está realizando. De esta forma podemos comprobar si un perfil tiene permiso para acceder o no a la pantalla correspondiente.

```

/**
 * This class manages the access to different views and controllers for different admin users.
 */
class CBackadmprofilesController extends CAppController {

    public function CBackadmprofilesController()
    {
        parent::CAppController();
        $this->setType( C_BACKEND );
        $this->_someModels = array("CAdmin", "CUser");
        $this->_someControllers = array("CCsvsController");
    }
}

```

```

    $this->_sLayout = "backlayout";
}

/**
 * This function controls if an admin can access to a view.
 * @param type $p_sControllerTarget The name of the controller to access
 * @param type $p_sViewTarget The name of the view to access
 * @return type If has access it will return true otherwise it will return false
 */
public function hasAccess($p_sControllerTarget, $p_sViewTarget)
{
    try
    {
        $bReturn = true;
        $oSession = new CAppSession();
        $iIdAdmin = $oSession->__get("idadmin");
        //if he is an admin
        if ($iIdAdmin != false)
        {
            $someProfile = $oSession->__get("someProfile");

            if (count($someProfile)>0)
            {
                if (isset($someProfile[0][$p_sControllerTarget]) && ($someProfile[0][$p_sControllerTarget] >
C_PROFILE_ADM_NO))
                {
                    if (($p_sViewTarget == "edit") || ($p_sViewTarget == "delete"))
                    {
                        if ($someProfile[0][$p_sControllerTarget] <= C_PROFILE_ADM_R) //only to edit
                        {
                            $bReturn = false;
                        }
                    }
                }
            }
        }
    }
}
catch(CException $e)

```

```

    {
        throw $e;
    }
    return $bReturn;
}
}

```

5.1.4.2 Área colaborativa.

El área colaborativa es la encargada de gestionar el chat y el timeline de acciones/peticiones que permiten colaborar a los diferentes actores del sistema. El controlador encargado de gestionar todas estas acciones es el controlador Ctimelinecontroller. A nivel de modelo se encargará la clase Ctimeline. En este caso esta clase de modelo se encargará de gestionar el acceso a la tabla timelines y chat. Aunque se podría haber decidido utilizar otra clase de modelo para el acceso chat se considera que inicialmente puede ser suficiente con una única clase que se encargue de gestionar a las dos tablas.

La clase Ctimelinescontroller está formada por una serie de métodos públicos:

```

class CTimelinesController extends CAppController {
    private $_iNumberItemsPerPage = 100;
    public function CTimelinesController(){
        parent::CAppController();
        $this->setType( C_BACKEND );
        $this->_sLayout = "backlayout";
        $this->_someModels = array("CTimeline","CUser","CCountrie", "CRegion", "CCitie", "CLanguage", "CAAdmin");
    }
    public function index()
    public function search()
    public function sendRequest($p_iAction, $p_iIdFrom, $p_iIdTo, $p_sExtra = "")
    public function sendAnswer($p_iAction, $p_iIdFrom, $p_iIdTo, $p_sExtra="")
    public function accept($p_iIdTimeline)
    public function reject($p_iIdTimeline)
    public function delete( $p_iId = null)
    public function chat()
    public function addmessage()
    public function getmessages()
}

```

De estos métodos, los que hacen referencia a las peticiones son sendRequest y sendAnswer que sirven como canal para establecer la comunicación de la petición de una acción. Por otro lado los métodos accept y reject son los que son ejecutados cuando un usuario realiza la acción de aceptar o rechazar una solicitud en el timeline. Además también se encargan de enviar las notificaciones via email.

Veamos el código de alguno de ellos:

```

public function accept($p_idTimeline)
{
    try
    {
        $someTimelines = $this->_oTimeline->getTimeLineById($p_idTimeline);
        if (count($someTimelines)>0)
        {
            $someToId = explode(":",$someTimelines[0]["text"]);
            $iIdUser = trim($someToId[2]);
            $someUser = $this->_oUser->get_someUsersByIdWithoutDelete($iIdUser);
            if ($someTimelines[0]["action"] == C_TIMELINE_CREATE_USER)
            {
                //change deleted to 1 the user and adds to admins_users
                $this->_oUser->activateDeactivate($iIdUser,0);
                $this->_oUser->addUserToAdmin($someTimelines[0]["admins_source_id"],$iIdUser);
            }
            elseif ($someTimelines[0]["action"] == C_TIMELINE_MODIFY_USER)
            {
                $this->_oUser->activateDeactivate($iIdUser,0);
            }
            elseif ($someTimelines[0]["action"] == C_TIMELINE_DELETE_USER)
            {
                //change deleted to 1 the user and adds to admins_users
                $this->_oUser->activateDeactivate($iIdUser,1);
                $this->_oUser->deleteUserToAdmin($someTimelines[0]["admins_source_id"],$iIdUser);
            }
            elseif ($someTimelines[0]["action"] == C_TIMELINE_ACTIVATE_USER)
            {
                //change deleted to 1 the user and adds to admins_users
                $this->_oUser->activateDeactivate($iIdUser,0);
            }
            elseif ($someTimelines[0]["action"] == C_TIMELINE_DEACTIVATE_USER)
            {
                $this->_oUser->activateDeactivate($iIdUser,1);
            }
            //send notification
            $this->sendAnswer(C_TIMELINE_STATE_ACCEPTED,$someTimelines[0]["admins_target_id"],
                $someTimelines[0]["admins_source_id"], buildEndMessage($iIdUser, $someUser[0]["name"]));
            $this->_oTimeline->changeState($p_idTimeline, C_TIMELINE_STATE_ACCEPTED);
        }
    }
}

```

```

    redirect($this->link ("index", "timelines" ), __("Action Accepted",$this->_sLanguage,true), C_MESSAGE_OK,true);
    die();
}
catch(CException $e)
{
    throw $e;
}

```

Dentro de cada una de las funciones se evalúa a que tipo de notificación TIMELINE corresponde la acción y dependiendo de cada una de ellas se ejecutará el código correspondiente.

5.1.4.3 Área de Chat.

La aplicación incorpora un chat a través del cual se pueden enviar mensajes entre los diferentes usuarios que hay en la aplicación. En esta primera versión se ha creado un chat genérico de modo que no se elige un destinatario sino que es como si se tratara de una sala común donde todos los mensajes que se escriban serán visualizados por todos los usuarios.

Lo métodos `addmessage` y `getmessages` de la clase `CtimelineController` son los encargados de escribir en la tabla de chat. Una vez escritos en la tabla con ayuda de `ajax` y `jquery` se van mostrando los mensajes correspondientes a los usuarios que están conectados en la pantalla de chat. A continuación mostramos el código para añadir mensajes y visualizarlos.

```

public function addmessage()
{
    try
    {
        $this->_sLayout = "ajax";
        $someParameters = $this->get_someParameters();
        $sText = $someParameters[0];
        $oSession = new CAppSession();
        $iIdAdmin = $oSession->__get("idadmin");
        $someToInsert = array();
        $someToInsert["chat"]["text"] = $sText;
        $someToInsert["chat"]["admins_source_id"] = $iIdAdmin;
        $someToInsert["chat"]["deleted"] = 0;
        $iResult = $this->_oTimeline->insert($someToInsert,false,false);
        if ($iResult == C_WARNING_INSERTEDOK)
        {
            $iIdLastInserted = $this->_oTimeline->get_iIdLastInserted();
            if ($oSession->__get("idLineChat") == false)
            {
                $oSession->__set("idLineChat",$iIdLastInserted);
            }
        }
    }
}

```

```

    }
}

$this->_oView->add("iReturn",$iResult);
}
catch(CException $e)
{
    throw $e;
}
}

public function getmessages()
{
    try
    {
        $oSession = new CAppSession();
        $this->_sLayout = "ajax";
        $sHtml = "";
        if ($oSession->__get("idLineChat") != false)
        {
            $this->_oTimeline->select(
                array(
                    'p_sColumns' => '*',
                    "p_sTable" => "chat",
                    "p_sWhere" => "id >= ".$oSession->__get("idLineChat"),

                )
            );
            $someData = $this->_oTimeline->get_someData();
            $sHtml = "";
            foreach($someData as $oData)
            {
                $someAdmins = $this->_oAdmin->getSomeAdminsById($oData["admins_source_id"]);

                $sHtml .= "<p>".$someAdmins[0]["company_name"].": ".$oData["text"]."</p>";
            }
        }
        $this->_oView->add("sHtml",$sHtml);
    }
    catch(CException $e)

```

```

    {
        throw $e;
    }
}

```

Lo que se hace en el lado del cliente es emitir una petición ajax cada x segundos para refrescar la zona del chat. Esto se realiza en el archivo chat.js que se carga al entrar en la pantalla de chat. Esto se realizará mediante tecnología jquery.

```

$(document).ready(function(){

function loadLog()
{
    var oldscrollHeight = $("#chatbox").attr("scrollHeight") - 20;
    var sMessages = behaviours.callController("timelines", "getmessages", $("#usermsg").val());

    //Auto-scroll
    $("#chatbox").html(sMessages);
    var newscrollHeight = $("#chatbox").attr("scrollHeight") - 20;
    if(newscrollHeight > oldscrollHeight){
        $("#chatbox").animate({ scrollTop: newscrollHeight }, 'normal');
    }
}

$("#submitmsg").click(function(){
    var sReturnValue = behaviours.callController("timelines", "addmessage", $("#usermsg").val());
    $("#usermsg").attr("value", "");
    return false;
});

setInterval (loadLog, 2500);
});

```

5.1.4.4 Área de administración de usuarios.

Desde el perfil de administrador se pueden gestionar diferentes tipos de clientes. Se pueden crear, editar, eliminar, exportar, etc. La clase CbackusersController permite gestionar todas estas acciones y dota a la aplicación de una interfaz para realizarlas de forma cómoda.

La clase se compone de los siguientes métodos.

```
class CBackusersController extends CAppController {
    private $_iNumberItemsPerPage = 100;
    public function CBackusersController(){
        parent::CAppController();
        $this->setType( C_BACKEND );
        $this->_someModels = array("CUser","CCountrie", "CRegion", "CCitie", "CLanguage");
        $this->_someControllers = array("CCsvsController");
        $this->_sLayout = "backlayout";
        $this->someCss[] = "backusers.css";
        $this->_oLayout->add('someCss',$this->someCss);
    }
    public function index()
    public function edit($iId = null)
    public function add()
    public function delete()
    public function search()
    public function changestate($p_iId, $p_bRequest = 0)
    public function exportesv()
    public function accessaccount($p_iIdUser)
}
```

A continuación veremos el código de algunas de las funciones que se utilizan en esta clase y que se han considerado importantes en la ejecución del código.

En el **método delete** hay que tener en cuenta que cuando se realiza la acción podrá estar realizándola una administrador o otro usuario de otro perfil. En este segundo caso, a diferencia de como ocurre cuando lo ejecuta un usuario de tipo administrador, se solicitará la acción de eliminarlo, no se eliminará directamente. Podemos verlo en las líneas en negrita:

```
public function delete( $p_iId = null, $p_bRequest = 0 ) {
    try {
        //First we have to call the index View to create variables that will be used in the View
        $this->index();
        $oSession = new CAppSession();
        //if you want to add extra operations to this method, you will have to do it here
        //first we have to delete
        if ( !is_null($p_iId) ) {
```

```

if ($p_bRequest == 0)
{
    //admin

    $iReturnedValue = $this->_oUser->delete($this->_oUser->get_sTable(), 'id = '.$p_iId);
}
else
{
    //other

    $someUser = $this->_oUser->get_someUsersByIdWithoutDelete($p_iId);
    //SEND request
    $oTimeLineController = $this->getUsedClass("CTimelinesController");
    $iReturnedValue = $oTimeLineController->sendRequest(C_TIMELINE_DELETE_USER, $oSession->__get("idadmin"), C_ROOT_ADMIN,
        buildEndMessage($someUser[0]["id"], $someUser[0]["name"]));
    $iReturnedValue=C_WARNING_DELETEOK;
}
//if the delete operation success, we will show a message with 'deleted correctly'
if ($iReturnedValue == C_WARNING_DELETEOK) {
    $this->_oLayout->add('sTextMessage', __("Operation executed successfully",$this->_sLanguage, true));
    $this->_oLayout->add('sTypeMessage',C_MESSAGE_OK);
}
else {
    $this->_oLayout->add('sTextMessage', __("Operation NOT executed successffuly",$this->_sLanguage, true));
    $this->_oLayout->add('sTypeMessage',C_MESSAGE_ERROR);
}

}
//if you want to add extra operations to this method, you will have to do it here
$this->_oView->set_sView('index.php');
}
catch(CException $e) {
    throw $e;
}
}

```

El **método index** de la clase se encarga de mostrar los listados de los clientes:

```

public function index() {
    try {
        $someParts = $this->_oUser->getLanguageIdByLanguageCode($this->_sLanguage);
        $iCodeLang = $someParts[0]['id'];
    }
}

```

```

$oSesion = new CAppSession();
//it gets the profile to be evaluated
$someProfile = $oSesion->__get("someProfile");
$iAdminId = $oSesion->__get("iadmin");
if ($someProfile[0]["id"] == C_PROFILE_ROOT) //only for roots
{
    $someValues = array(
        'p_sColumns' => 'u.*, c.name as country',
        'p_sTable' => "users u LEFT JOIN countries c ON u.countries_id = c.id",
        'p_sWhere' => "u.type = ".C_PROF_USER." and u.parent_users_id = -1 AND c.languages_id = $iCodeLang",
        'p_sOrder' => 'u.expiration_date asc',
        'p_sLimit' => $this->_iNumberItemsPerPage);
}
else
{
    $someValues = array(
        'p_sColumns' => 'u.*, c.name as country',
        'p_sTable' => "users u LEFT JOIN countries c ON u.countries_id = c.id",
        'p_sWhere' => "(u.id IN (SELECT au.users_id FROM admins_users as au WHERE au.admins_id = $iAdminId))
AND (u.type = ".C_PROF_USER." and u.parent_users_id = -1 AND c.languages_id = $iCodeLang)",
        'p_sOrder' => 'u.expiration_date asc',
        'p_sLimit' => $this->_iNumberItemsPerPage);
}

$oSesion->__setArray("someToCsv",$someValues);
$someData = $this->paginate('CUser', 'index', 'backusers',$someValues);
$sLastInserted = 0;
if ( isset ( $this->_someParameters[0] ) )
    $sLastInserted = $this->_someParameters[0];

$someColumns = $this->_oUser->describeColumns('users',true);
$this->_oView->add('someColumns',$someColumns);
$this->_oView->add('someData',$someData);

//$this->_oView->add('sLastInserted',$sLastInserted);

//Do Global configurations
$this->doGlobalConfiguration( array(
    'p_iMenu' => C_HEADER_MENU_TYPE_SIMPLE_GREEN,
    'p_sTitleMenu' => "Clients",
    'p_sImageMenu' => "global/submenu/centericon.png",
    'p_sButton' => C_HEADER_MAIN_MENU_DASHBOARD,

```

```
"p_iLevel" => 0,  
"p_iName" => "Clients",  
"p_sController" => "backusers",  
"p_sMethod" => "index"  
));  
  
}  
catch(CException $e) {  
    throw $e;  
}  
  
}
```

Igual que en el caso del método delete en este caso también se pueden diferenciar los listados de los administradores a los colaboradores o partners.

5.1.5 Instalación y configuración.

El código fuente suministrado es el mínimo necesario para que funcione correctamente el software. Se ha prescindido de módulos que no están relacionados con el proyecto. Los archivos incluidos que no forman parte del core del framework utilizado han sido:

Controladores

/app/controllers/CBackadminsController.php
/app/controllers/CBackusersController.php
/app/plugins/logins/CLoginsController.php
/app/controllers/CBackadmprofilesController.php
/app/controllers/CBackpaymentsController.php
/app/controllers/CTimelineController.php
/app/plugins/csvs/CCsvsController.php

Modelos

/app/models/CAdmin.php
/app/models/CAdmprofile.php
/app/models/CCitie.php
/app/models/CRegion.php
app/models/CCountry.php
/app/models/CLanguage.php
/app/models/CUser.php
/app/models/CTimeline.php
/app/plugins/csvs/models/CCsv.php
/app/models/CPayment.php

Vistas

/app/plugins/csvs/views/export.php
/app/plugins/csvs/views/index.php

/app/views/backend/backusers/add.php
/app/views/backend/backusers/edit.php
/app/views/backend/backusers/exportcsv.php
/app/views/backend/backusers/index.php

/app/views/backend/backadmins/add.php

```
/app/views/backend/backadmins/edit.php
/app/views/backend/backadmins/exportcsv.php
/app/views/backend/backadmins/index.php
```

```
/app/plugins/logins/views/backend.php
```

```
/app/views/backend/backpayments/add.php
/app/views/backend/backpayments/edit.php
/app/views/backend/backpayments/exportcsv.php
/app/views/backend/backpayments/index.php
/app/views/backend/backpayments/printpayment.php
```

```
/app/views/backend/timelines/addmessage.php
/app/views/backend/timelines/chat.php
/app/views/backend/timelines/getmessages.php
/app/views/backend/timelines/index.php
```

Instalación

Para realizar la instalación de la aplicación se necesita una versión de PHP 5.2 o superior, MySQL 5.0 o superior y un servidor Apache 2.2. Adicionalmente se tendrán que instalar los módulos lib-curl, mod-rewrite.

Una vez configurado el servidor de base de datos y servidor apache se procederá a la instalación. Para ello simplemente tendremos que copiar la carpeta con la aplicación al directorio de publicación de apache (dependerá de la distribución de linux en "ubuntu" /var/www/).

Una vez copiada la aplicación tendremos que editar el archivo de configuración situado en la ruta (dentro de la carpeta de la aplicación) /app/config/app_config.php

Este archivo contiene los parámetros de configuración de la aplicación. Simplemente tendremos que modificar las constantes:

```
define("C_ROOT_KEY", "/proyecto");
define("C_ROOT_DOMAIN", "localhost");

define("C_BDA_USER", "root");
define("C_BDA_PASSWORD", "");
define("C_BDA_HOST", "localhost");
define("C_BDA_NAME", "proyecto");
```

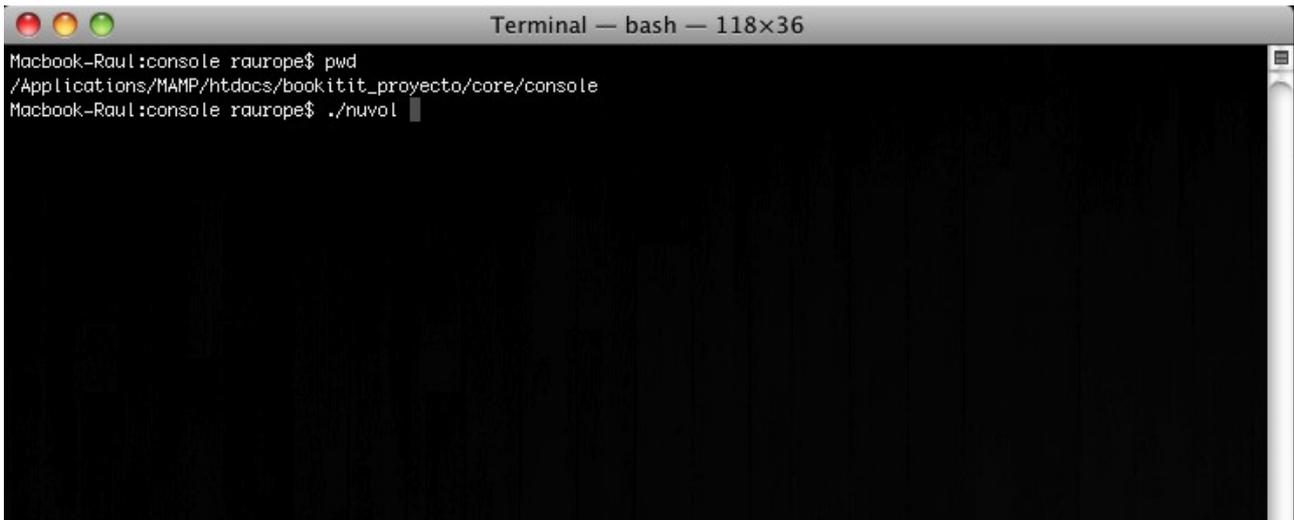
Las constantes C_ROOT_KEY y C_ROOT_DOMAIN sirven para indicar donde está situada la aplicación. Por ejemplo si se encuentra en la raíz del dominio www.prueba.com la configuración

sería:

```
define("C_ROOT_KEY", "");
define("C_ROOT_DOMAIN", "www.prueba.com");
```

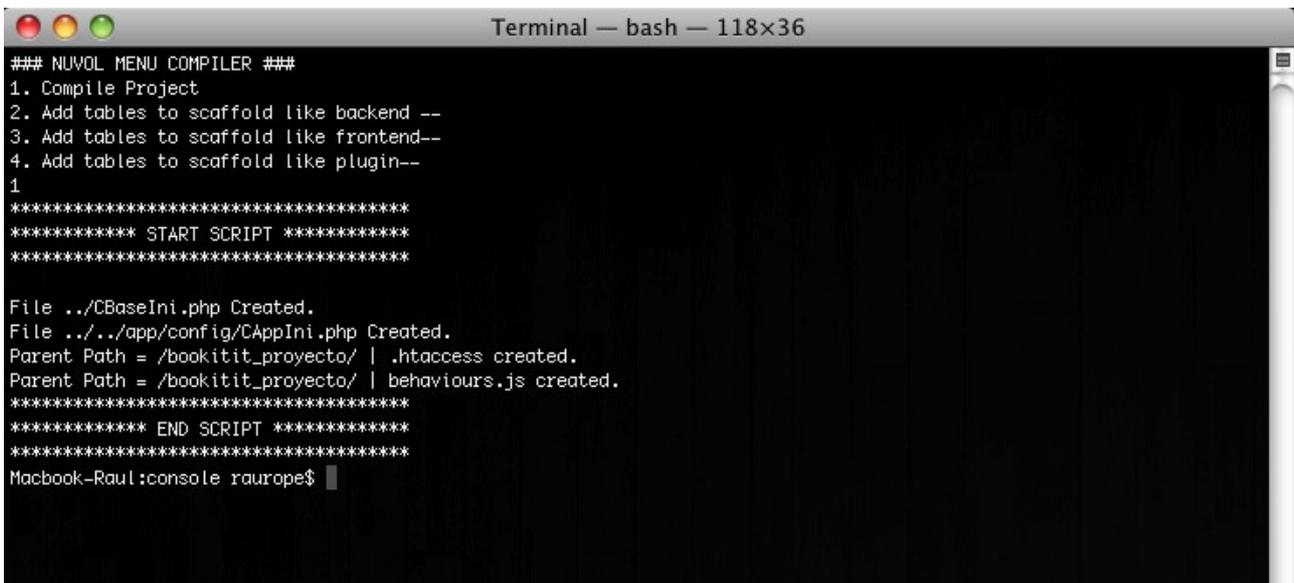
El resto de variables servirán para configurar la base de datos que hayamos creado y que permitirá a la aplicación acceder a la base de datos.

Una vez configurados todos estos parámetros tendremos que acceder a un terminal o consola y regenerar la aplicación. Para ello accedemos al directorio `/core/console` y ejecutamos el `.bat` o `.sh` `nuvol`



```
Terminal — bash — 118x36
Macbook-Raul:console raurope$ pwd
/Applications/MAMP/htdocs/bookitit_proyecto/core/console
Macbook-Raul:console raurope$ ./nuvol
```

Una vez hayamos ejecutado el comando `nuvol`, podremos regenerar el proyecto pulsando la opción 1 y enter.



```
Terminal — bash — 118x36
### NUVOL MENU COMPILER ###
1. Compile Project
2. Add tables to scaffold like backend --
3. Add tables to scaffold like frontend--
4. Add tables to scaffold like plugin--
1
*****
***** START SCRIPT *****
*****
File ../CBaseIni.php Created.
File ../../app/config/CAppIni.php Created.
Parent Path = /bookitit_proyecto/ | .htaccess created.
Parent Path = /bookitit_proyecto/ | behaviours.js created.
*****
***** END SCRIPT *****
*****
Macbook-Raul:console raurope$
```

En ese momento se regenerarán los archivos base necesarios con la configuración que hemos introducido y tendremos la aplicación lista para funcionar en nuestro directorio de publicación.

5.1.6 Pruebas de estabilidad y carga.



Para la elaboración de esta sección se ha empleado la herramienta de Test Stress Jmeter.

Jmeter es una herramienta open source desarrollada en Java que permite realizar pruebas sobre una aplicación web simulando varias peticiones concurrentes al servidor de forma que podemos ver como reacciona la aplicación mediante gráficas y tablas de resultados. De esta forma podemos ver como reaccionará la aplicación ante situaciones de máximo tráfico y así saber a partir de cuando el servidor podría dejar de atender peticiones o empezar un proceso de ralentización.

Aunque en la mayoría de proyectos no se tienen en cuenta este tipo de pruebas (ni otras), es muy conveniente realizarlas ya que de este modo se podría saber a partir de que tráfico habría que escalar a otro servidor o optimizar el código fuente antes de que la situación llegara a ser crítica. En resumen se trataría de adelantarnos, dependiendo del tráfico generado actualmente, a una situación en la que por sorpresa dejaran al servidor y la aplicación sin funcionamiento.

Ejemplo de pantalla de Jmeter

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes
1	20:37:52.980	Grupo de Hilos 1-1	/uoc/backusers/add	1716	▲	1860
2	20:37:53.032	Grupo de Hilos 1-2	/uoc/backusers/add	2056	▲	1860
3	20:37:53.082	Grupo de Hilos 1-3	/uoc/backusers/add	2329	▲	1860
4	20:37:53.133	Grupo de Hilos 1-4	/uoc/backusers/add	2474	▲	1860
5	20:37:53.183	Grupo de Hilos 1-5	/uoc/backusers/add	2620	▲	1860
6	20:37:53.234	Grupo de Hilos 1-6	/uoc/backusers/add	2674	▲	1860
7	20:37:53.291	Grupo de Hilos 1-7	/uoc/backusers/add	2824	▲	1860
8	20:37:53.342	Grupo de Hilos 1-8	/uoc/backusers/add	2885	▲	1860
9	20:37:53.393	Grupo de Hilos 1-9	/uoc/backusers/add	2890	▲	1860
10	20:37:53.442	Grupo de Hilos 1-10	/uoc/backusers/add	3003	▲	1860
11	20:37:53.493	Grupo de Hilos 1-11	/uoc/backusers/add	3510	▲	1860
12	20:37:54.714	Grupo de Hilos 1-35	/uoc/backusers/add	4181	▲	1859
13	20:37:53.594	Grupo de Hilos 1-13	/uoc/backusers/add	5798	▲	1860
14	20:37:53.544	Grupo de Hilos 1-12	/uoc/backusers/add	5928	▲	1860
15	20:37:55.116	Grupo de Hilos 1-43	/uoc/backusers/add	4759	▲	1859
16	20:37:55.418	Grupo de Hilos 1-49	/uoc/backusers/add	5557	▲	1859
17	20:37:55.619	Grupo de Hilos 1-53	/uoc/backusers/add	6127	▲	1859
18	20:37:55.820	Grupo de Hilos 1-57	/uoc/backusers/add	6351	▲	1859
19	20:37:55.919	Grupo de Hilos 1-59	/uoc/backusers/add	6457	▲	1859
20	20:37:53.645	Grupo de Hilos 1-14	/uoc/backusers/add	9123	▲	1860
21	20:37:53.695	Grupo de Hilos 1-15	/uoc/backusers/add	9090	▲	1860
22	20:37:53.745	Grupo de Hilos 1-16	/uoc/backusers/add	9436	▲	1860
23	20:37:53.795	Grupo de Hilos 1-17	/uoc/backusers/add	9420	▲	1860
24	20:37:53.846	Grupo de Hilos 1-18	/uoc/backusers/add	11378	▲	1860
25	20:37:53.946	Grupo de Hilos 1-20	/uoc/backusers/add	11413	▲	1860
26	20:37:53.896	Grupo de Hilos 1-19	/uoc/backusers/add	11536	▲	1860

Scroll automatically?
 Child samples?
 No. de Muestras 60
 Última Muestra 12285
 Media 9474
 Desviación 3993

```

2012/05/29 20:38:08 INFO - jmeter.engine.StandardJMeterEngine: Ending thread Grupo de Hilos 1-54
2012/05/29 20:38:08 INFO - jmeter.threads.JMeterThread: Thread finished: Grupo de Hilos 1-58
2012/05/29 20:38:08 INFO - jmeter.engine.StandardJMeterEngine: Ending thread Grupo de Hilos 1-58
2012/05/29 20:38:08 INFO - jmeter.threads.JMeterThread: Thread finished: Grupo de Hilos 1-56
2012/05/29 20:38:08 INFO - jmeter.engine.StandardJMeterEngine: Ending thread Grupo de Hilos 1-56
2012/05/29 20:38:08 INFO - jmeter.engine.StandardJMeterEngine: Notifying test listeners of end of test
2012/05/29 20:38:08 INFO - jmeter.gui.util.JMeterMenuBar: setRunning(false,"local")
2012/05/29 20:38:08 INFO - jmeter.engine.StandardJMeterEngine: Test has ended on host null
2012/05/29 20:40:11 INFO - jmeter.services.FileServer: Set new base=/Applications/apache-jmeter-2.7/pruebasraul/
  
```

A continuación mostraremos las pruebas que se realizaron en la aplicación. El servidor empleado para la ejecución de las pruebas de test stress fué un AMD ATHLON 900Mhz con 1.5GB de RAM.

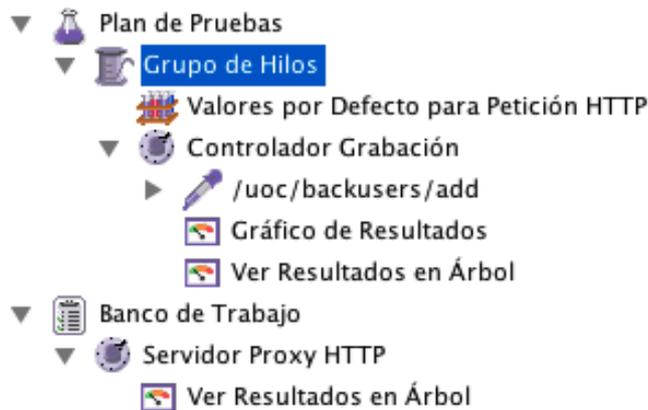
Cada prueba fué ejecutada con un número simulado de usuarios simultáneos: 1 petición por segundo, 10 peticiones por segundo y 20 peticiones por segundo.

Pruebas realizadas

-Validación de administrador.

-Inserción de cliente desde un Administrador.

-Solicitud de alta de cliente desde un Partner.



Para la ejecución de las pruebas se configuró la herramienta Jmeter con la utilización de un proxy. Este proxy es utilizado por Jmeter para almacenar las acciones que realiza el usuario y luego poder reproducirlos de forma virtual con el número que queramos de hilos.

En Jmeter existen 2 valores que hay que configurar en la sección de “Grupo de Hilos”. Cuando creas un grupo de hilos hay que establecer el valor para dos variables importantes: número de hilos y periodo de subida.

Si clicamos en esta sección aparecerán las siguientes casillas:

En esta sección aparecen 2 campos importantes, el número de hilos y el periodo de subida. El número de hilos será equivalente al número de usuarios que repetirán la acción y el periodo de subida nos servirá para indicar el número de peticiones por segundo que harán estos usuarios, es decir aunque sean 60 usuarios sólo habrá un número de estos usuarios que harán las peticiones dentro del mismo segundo.

Para el cálculo de estos valores se utiliza la siguiente fórmula:

$$\frac{\text{número de hilos}}{\text{periodo de subida}} = \text{Número de hilos por segundo}$$

Lo que se recomienda es iniciar el número de peticiones a 1 por segundo. Para obtener este resultado utilizamos la fórmula reemplazando por los siguientes valores:

$$\frac{60 \text{ hilos}}{60 \text{ periodo de subida}} = 1 \text{ hilo por segundo}$$

Este será el primer tipo de prueba que haremos en cualquier caso y simularía a un usuario haciendo realizando una petición en cada segundo.

El resto de pruebas serán para 10 peticiones por segundo y 20 peticiones por segundo. Las fórmulas quedarán de la siguiente forma:

$$\frac{60 \text{ hilos}}{6 \text{ periodo de subida}} = 10 \text{ hilos por segundo} \quad \frac{60 \text{ hilos}}{3 \text{ periodo de subida}} = 20 \text{ hilos por segundo}$$

TEST: VALIDACIÓN DE ADMINISTRADOR.

Extracto del resultado de 1 petición.

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes
1	21:36:29.826	Grupo de Hilos 1-1	/uoc/logins/backend	392		2866
2	21:36:30.219	Grupo de Hilos 1-1	/uoc/logins/backend	429		2391
3	21:36:30.649	Grupo de Hilos 1-1	/uoc/backusers/index	237		1859
4	21:36:30.828	Grupo de Hilos 1-2	/uoc/logins/backend	246		2866
5	21:36:31.075	Grupo de Hilos 1-2	/uoc/logins/backend	459		2391
6	21:36:31.535	Grupo de Hilos 1-2	/uoc/backusers/index	250		1859
7	21:36:31.828	Grupo de Hilos 1-3	/uoc/logins/backend	204		2865
8	21:36:32.033	Grupo de Hilos 1-3	/uoc/logins/backend	454		2391
9	21:36:32.489	Grupo de Hilos 1-3	/uoc/backusers/index	253		1859
10	21:36:32.830	Grupo de Hilos 1-4	/uoc/logins/backend	197		2865
11	21:36:33.032	Grupo de Hilos 1-4	/uoc/logins/backend	436		2391
12	21:36:33.470	Grupo de Hilos 1-4	/uoc/backusers/index	238		1859
13	21:36:33.830	Grupo de Hilos 1-5	/uoc/logins/backend	203		2865
14	21:36:34.040	Grupo de Hilos 1-5	/uoc/logins/backend	462		2391
15	21:36:34.504	Grupo de Hilos 1-5	/uoc/backusers/index	264		1859
16	21:36:34.831	Grupo de Hilos 1-6	/uoc/logins/backend	217		2865
17	21:36:35.049	Grupo de Hilos 1-6	/uoc/logins/backend	439		2391
18	21:36:35.489	Grupo de Hilos 1-6	/uoc/backusers/index	244		1859
19	21:36:35.831	Grupo de Hilos 1-7	/uoc/logins/backend	218		2865
20	21:36:36.050	Grupo de Hilos 1-7	/uoc/logins/backend	450		2391
21	21:36:36.501	Grupo de Hilos 1-7	/uoc/backusers/index	240		1859
22	21:36:36.830	Grupo de Hilos 1-8	/uoc/logins/backend	199		2865
23	21:36:37.032	Grupo de Hilos 1-8	/uoc/logins/backend	449		2391
24	21:36:37.482	Grupo de Hilos 1-8	/uoc/backusers/index	249		1859
25	21:36:37.831	Grupo de Hilos 1-9	/uoc/logins/backend	214		2865
26	21:36:38.046	Grupo de Hilos 1-9	/uoc/logins/backend	446		2391

Podemos observar que el tiempo de las líneas en azul es de un segundo. Hay que tener en cuenta que en la muestra 3 el tiempo de comienzo si se suma el tiempo de muestra podemos ver que se aproxima aún más a 1 segundo. Además sólo se hace una petición y que los tiempos de muestra en milisegundos son muy bajos para cada una de las peticiones. **En este caso podríamos decir que no habría problemas para el servidor en atender este tipo de peticiones.**

Extracto del resultado de 10 peticiones.

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes
1	22:00:44.799	Grupo de Hilos 1-1	/uoc/logins/backend	1030		2866
2	22:00:44.900	Grupo de Hilos 1-2	/uoc/logins/backend	1177		2866
3	22:00:45.000	Grupo de Hilos 1-3	/uoc/logins/backend	1294		2866
4	22:00:45.101	Grupo de Hilos 1-4	/uoc/logins/backend	1532		2866
5	22:00:45.208	Grupo de Hilos 1-5	/uoc/logins/backend	1857		2866
6	22:00:45.309	Grupo de Hilos 1-6	/uoc/logins/backend	2264		2866
7	22:00:45.410	Grupo de Hilos 1-7	/uoc/logins/backend	2616		2866
8	22:00:45.510	Grupo de Hilos 1-8	/uoc/logins/backend	2931		2866
9	22:00:45.611	Grupo de Hilos 1-9	/uoc/logins/backend	3180		2866
10	22:00:45.712	Grupo de Hilos 1-10	/uoc/logins/backend	3427		2866
11	22:00:45.812	Grupo de Hilos 1-11	/uoc/logins/backend	3869		2866
12	22:00:46.014	Grupo de Hilos 1-13	/uoc/logins/backend	5123		2866
13	22:00:45.912	Grupo de Hilos 1-12	/uoc/logins/backend	5262		2866
14	22:00:48.027	Grupo de Hilos 1-33	/uoc/logins/backend	4746		2865
15	22:00:45.830	Grupo de Hilos 1-1	/uoc/logins/backend	8456		2391
16	22:00:46.217	Grupo de Hilos 1-15	/uoc/logins/backend	9264		2866
17	22:00:46.315	Grupo de Hilos 1-16	/uoc/logins/backend	9318		2866
18	22:00:46.113	Grupo de Hilos 1-14	/uoc/logins/backend	9607		2866
19	22:00:46.415	Grupo de Hilos 1-17	/uoc/logins/backend	9503		2866
20	22:00:46.079	Grupo de Hilos 1-2	/uoc/logins/backend	10118		2391
21	22:00:46.295	Grupo de Hilos 1-3	/uoc/logins/backend	11906		2391
22	22:00:46.517	Grupo de Hilos 1-18	/uoc/logins/backend	12782		2866
23	22:00:46.719	Grupo de Hilos 1-20	/uoc/logins/backend	12618		2866
24	22:00:46.821	Grupo de Hilos 1-21	/uoc/logins/backend	12557		2866
25	22:00:46.922	Grupo de Hilos 1-22	/uoc/logins/backend	12467		2866
26	22:00:47.123	Grupo de Hilos 1-24	/uoc/logins/backend	12402		2866

En este caso podemos ver que con 10 peticiones dentro del mismo segundo el servidor tarda bastante más en responder. Por ejemplo en la muestra 10 podemos ver que ya lleva 3427 ms consumidos y aún no ha respondido ya que debe ejecutar index.

Extracto del resultado de 20 peticiones.

El caso de 20 peticiones se va a mostrar pero veremos que los tiempos pasan a ser demasiado exagerados. No se puede tener a un usuarios más de 5-8 segundos esperando ya que pensaría que la aplicación deja de responder. También hay que tener en cuenta que estamos hablando de 20 peticiones dentro de un mismo segundo, eso supondría un tráfico muy elevado al final del día, en estas condiciones habría que tener un servidor más potente.

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes
1	22:07:01.110	Grupo de Hilos 1-1	/uoc/backusers/add	1677		1860
2	22:07:01.162	Grupo de Hilos 1-2	/uoc/backusers/add	2081		1860
3	22:07:01.220	Grupo de Hilos 1-3	/uoc/backusers/add	2497		1860
4	22:07:01.263	Grupo de Hilos 1-4	/uoc/backusers/add	2666		1860
5	22:07:01.314	Grupo de Hilos 1-5	/uoc/backusers/add	2765		1860
6	22:07:01.365	Grupo de Hilos 1-6	/uoc/backusers/add	2954		1860
7	22:07:01.418	Grupo de Hilos 1-7	/uoc/backusers/add	3016		1860
8	22:07:01.469	Grupo de Hilos 1-8	/uoc/backusers/add	2988		1860
9	22:07:01.520	Grupo de Hilos 1-9	/uoc/backusers/add	3029		1860
10	22:07:01.571	Grupo de Hilos 1-10	/uoc/backusers/add	3178		1860
11	22:07:01.621	Grupo de Hilos 1-11	/uoc/backusers/add	3665		1860
12	22:07:02.804	Grupo de Hilos 1-34	/uoc/backusers/add	4526		1859
13	22:07:01.672	Grupo de Hilos 1-12	/uoc/backusers/add	5727		1860
14	22:07:01.723	Grupo de Hilos 1-13	/uoc/backusers/add	5686		1860
15	22:07:03.257	Grupo de Hilos 1-43	/uoc/backusers/add	5624		1859
16	22:07:03.766	Grupo de Hilos 1-53	/uoc/backusers/add	6510		1859
17	22:07:01.873	Grupo de Hilos 1-16	/uoc/backusers/add	8857		1860
18	22:07:01.924	Grupo de Hilos 1-17	/uoc/backusers/add	8835		1860
19	22:07:01.823	Grupo de Hilos 1-15	/uoc/backusers/add	8963		1860
20	22:07:03.968	Grupo de Hilos 1-57	/uoc/backusers/add	7079		1859
21	22:07:01.773	Grupo de Hilos 1-14	/uoc/backusers/add	9308		1860
22	22:07:04.123	Grupo de Hilos 1-60	/uoc/backusers/add	7038		1859
23	22:07:02.108	Grupo de Hilos 1-20	/uoc/backusers/add	11185		1860
24	22:07:02.108	Grupo de Hilos 1-19	/uoc/backusers/add	11225		1860
25	22:07:02.136	Grupo de Hilos 1-21	/uoc/backusers/add	11200		1860
26	22:07:01.975	Grupo de Hilos 1-18	/uoc/backusers/add	11387		1860

Además en estas situaciones podemos ver que se cambia el orden de los hilos. Esto puede ser

debido a los altos tiempos de respuesta.

TEST: INSERCIÓN DE CLIENTE.

Extracto del resultado de 1 petición.

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes
1	21:46:25.232	Grupo de Hilos 1-1	/uoc/backusers/add	296		1860
2	21:46:26.234	Grupo de Hilos 1-2	/uoc/backusers/add	245		1859
3	21:46:27.235	Grupo de Hilos 1-3	/uoc/backusers/add	245		1859
4	21:46:28.235	Grupo de Hilos 1-4	/uoc/backusers/add	243		1859
5	21:46:29.236	Grupo de Hilos 1-5	/uoc/backusers/add	244		1859
6	21:46:30.236	Grupo de Hilos 1-6	/uoc/backusers/add	239		1859
7	21:46:31.236	Grupo de Hilos 1-7	/uoc/backusers/add	238		1859
8	21:46:32.236	Grupo de Hilos 1-8	/uoc/backusers/add	241		1859
9	21:46:33.236	Grupo de Hilos 1-9	/uoc/backusers/add	250		1859
10	21:46:34.239	Grupo de Hilos 1-10	/uoc/backusers/add	253		1859
11	21:46:35.238	Grupo de Hilos 1-11	/uoc/backusers/add	254		1859
12	21:46:36.247	Grupo de Hilos 1-12	/uoc/backusers/add	238		1859
13	21:46:37.247	Grupo de Hilos 1-13	/uoc/backusers/add	240		1859
14	21:46:38.248	Grupo de Hilos 1-14	/uoc/backusers/add	259		1859
15	21:46:39.249	Grupo de Hilos 1-15	/uoc/backusers/add	238		1859
16	21:46:40.248	Grupo de Hilos 1-16	/uoc/backusers/add	242		1859
17	21:46:41.249	Grupo de Hilos 1-17	/uoc/backusers/add	246		1859
18	21:46:42.250	Grupo de Hilos 1-18	/uoc/backusers/add	244		1859
19	21:46:43.249	Grupo de Hilos 1-19	/uoc/backusers/add	242		1859
20	21:46:44.250	Grupo de Hilos 1-20	/uoc/backusers/add	252		1859
21	21:46:45.251	Grupo de Hilos 1-21	/uoc/backusers/add	243		1859
22	21:46:46.251	Grupo de Hilos 1-22	/uoc/backusers/add	238		1859
23	21:46:47.252	Grupo de Hilos 1-23	/uoc/backusers/add	239		1859
24	21:46:48.252	Grupo de Hilos 1-24	/uoc/backusers/add	253		1859
25	21:46:49.252	Grupo de Hilos 1-25	/uoc/backusers/add	241		1859
26	21:46:50.253	Grupo de Hilos 1-26	/uoc/backusers/add	237		1859

Extracto del resultado de 10 peticiones.

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes
1	22:12:19.908	Grupo de Hilos 1-1	/uoc/backusers/add	520		1860
2	22:12:20.009	Grupo de Hilos 1-2	/uoc/backusers/add	1231		1860
3	22:12:20.109	Grupo de Hilos 1-3	/uoc/backusers/add	1711		1860
4	22:12:20.210	Grupo de Hilos 1-4	/uoc/backusers/add	1963		1860
5	22:12:20.310	Grupo de Hilos 1-5	/uoc/backusers/add	2153		1860
6	22:12:20.411	Grupo de Hilos 1-6	/uoc/backusers/add	2353		1860
7	22:12:20.511	Grupo de Hilos 1-7	/uoc/backusers/add	2349		1859
8	22:12:20.611	Grupo de Hilos 1-8	/uoc/backusers/add	2715		1860
9	22:12:20.712	Grupo de Hilos 1-9	/uoc/backusers/add	2844		1860
10	22:12:20.812	Grupo de Hilos 1-10	/uoc/backusers/add	2841		1860
11	22:12:20.913	Grupo de Hilos 1-11	/uoc/backusers/add	2958		1860
12	22:12:21.313	Grupo de Hilos 1-15	/uoc/backusers/add	3266		1859
13	22:12:21.012	Grupo de Hilos 1-12	/uoc/backusers/add	3767		1860
14	22:12:21.834	Grupo de Hilos 1-20	/uoc/backusers/add	4377		1859
15	22:12:22.236	Grupo de Hilos 1-24	/uoc/backusers/add	5181		1859
16	22:12:21.213	Grupo de Hilos 1-14	/uoc/backusers/add	6538		1860
17	22:12:21.114	Grupo de Hilos 1-13	/uoc/backusers/add	6660		1860
18	22:12:22.551	Grupo de Hilos 1-27	/uoc/backusers/add	6052		1859
19	22:12:22.840	Grupo de Hilos 1-30	/uoc/backusers/add	6611		1859
20	22:12:22.942	Grupo de Hilos 1-31	/uoc/backusers/add	6714		1859
21	22:12:21.515	Grupo de Hilos 1-17	/uoc/backusers/add	9463		1860
22	22:12:23.343	Grupo de Hilos 1-35	/uoc/backusers/add	7691		1859
23	22:12:21.616	Grupo de Hilos 1-18	/uoc/backusers/add	9481		1860
24	22:12:21.415	Grupo de Hilos 1-16	/uoc/backusers/add	9694		1860
25	22:12:21.731	Grupo de Hilos 1-19	/uoc/backusers/add	9602		1860
26	22:12:23.644	Grupo de Hilos 1-38	/uoc/backusers/add	7916		1859

En este resultado podemos ver que hasta 10 usuarios dentro del mismo segundo haciendo un insert respondería bastante bien, pero a partir de 12 o 13 si fuera en el siguiente segundo, el tiempo de respuesta empieza a subir de forma elevada hasta llegar a las 9.4 segundos de respuesta.

Extracto del resultado de 20 peticiones.

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes
1	22:13:46.801	Grupo de Hilos 1-1	/uoc/backusers/add	1485		1860
2	22:13:46.858	Grupo de Hilos 1-2	/uoc/backusers/add	1832		1860
3	22:13:46.909	Grupo de Hilos 1-3	/uoc/backusers/add	2583		1860
4	22:13:47.010	Grupo de Hilos 1-5	/uoc/backusers/add	2682		1860
5	22:13:46.959	Grupo de Hilos 1-4	/uoc/backusers/add	2748		1860
6	22:13:47.061	Grupo de Hilos 1-6	/uoc/backusers/add	2815		1860
7	22:13:47.111	Grupo de Hilos 1-7	/uoc/backusers/add	2893		1860
8	22:13:47.161	Grupo de Hilos 1-8	/uoc/backusers/add	3010		1860
9	22:13:47.211	Grupo de Hilos 1-9	/uoc/backusers/add	2962		1860
10	22:13:47.262	Grupo de Hilos 1-10	/uoc/backusers/add	2990		1860
11	22:13:47.312	Grupo de Hilos 1-11	/uoc/backusers/add	3473		1860
12	22:13:48.324	Grupo de Hilos 1-31	/uoc/backusers/add	3889		1859
13	22:13:47.370	Grupo de Hilos 1-12	/uoc/backusers/add	5759		1860
14	22:13:48.731	Grupo de Hilos 1-39	/uoc/backusers/add	4446		1859
15	22:13:47.414	Grupo de Hilos 1-13	/uoc/backusers/add	5790		1860
16	22:13:49.541	Grupo de Hilos 1-55	/uoc/backusers/add	6196		1859
17	22:13:49.742	Grupo de Hilos 1-59	/uoc/backusers/add	6639		1859
18	22:13:47.516	Grupo de Hilos 1-15	/uoc/backusers/add	8882		1860
19	22:13:47.465	Grupo de Hilos 1-14	/uoc/backusers/add	8995		1860
20	22:13:47.617	Grupo de Hilos 1-17	/uoc/backusers/add	8856		1860
21	22:13:47.566	Grupo de Hilos 1-16	/uoc/backusers/add	8990		1860
22	22:13:49.793	Grupo de Hilos 1-60	/uoc/backusers/add	6808		1859
23	22:13:47.769	Grupo de Hilos 1-20	/uoc/backusers/add	11083		1860
24	22:13:47.667	Grupo de Hilos 1-18	/uoc/backusers/add	11215		1860
25	22:13:47.819	Grupo de Hilos 1-21	/uoc/backusers/add	11091		1860
26	22:13:47.718	Grupo de Hilos 1-19	/uoc/backusers/add	11238		1860

TEST: SOLICITUD DE ALTA.**Extracto del resultado de 1 petición.**

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes
1	22:17:39.843	Grupo de Hilos 1-1	/uoc/backusers/add	754		1860
2	22:17:40.850	Grupo de Hilos 1-2	/uoc/backusers/add	249		1859
3	22:17:41.851	Grupo de Hilos 1-3	/uoc/backusers/add	243		1859
4	22:17:42.851	Grupo de Hilos 1-4	/uoc/backusers/add	244		1859
5	22:17:43.852	Grupo de Hilos 1-5	/uoc/backusers/add	240		1859
6	22:17:44.853	Grupo de Hilos 1-6	/uoc/backusers/add	239		1859
7	22:17:45.853	Grupo de Hilos 1-7	/uoc/backusers/add	255		1859
8	22:17:46.857	Grupo de Hilos 1-8	/uoc/backusers/add	247		1859
9	22:17:47.858	Grupo de Hilos 1-9	/uoc/backusers/add	243		1859
10	22:17:48.858	Grupo de Hilos 1-10	/uoc/backusers/add	240		1859
11	22:17:49.859	Grupo de Hilos 1-11	/uoc/backusers/add	240		1859
12	22:17:50.860	Grupo de Hilos 1-12	/uoc/backusers/add	241		1859
13	22:17:51.860	Grupo de Hilos 1-13	/uoc/backusers/add	240		1859
14	22:17:52.861	Grupo de Hilos 1-14	/uoc/backusers/add	241		1859
15	22:17:53.862	Grupo de Hilos 1-15	/uoc/backusers/add	260		1859
16	22:17:54.863	Grupo de Hilos 1-16	/uoc/backusers/add	238		1859
17	22:17:55.863	Grupo de Hilos 1-17	/uoc/backusers/add	241		1859
18	22:17:56.863	Grupo de Hilos 1-18	/uoc/backusers/add	241		1859
19	22:17:57.863	Grupo de Hilos 1-19	/uoc/backusers/add	241		1859
20	22:17:58.864	Grupo de Hilos 1-20	/uoc/backusers/add	244		1859
21	22:17:59.865	Grupo de Hilos 1-21	/uoc/backusers/add	239		1859
22	22:18:00.866	Grupo de Hilos 1-22	/uoc/backusers/add	239		1859
23	22:18:01.866	Grupo de Hilos 1-23	/uoc/backusers/add	244		1859
24	22:18:02.866	Grupo de Hilos 1-24	/uoc/backusers/add	241		1859
25	22:18:03.867	Grupo de Hilos 1-25	/uoc/backusers/add	243		1859
26	22:18:04.867	Grupo de Hilos 1-26	/uoc/backusers/add	242		1859

Extracto del resultado de 10 peticiones.

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes
1	22:22:22.267	Grupo de Hilos 1-1	/uoc/backusers/add	597		1860
2	22:22:22.368	Grupo de Hilos 1-2	/uoc/backusers/add	1269		1860
3	22:22:22.470	Grupo de Hilos 1-3	/uoc/backusers/add	1722		1860
4	22:22:22.571	Grupo de Hilos 1-4	/uoc/backusers/add	2047		1860
5	22:22:22.672	Grupo de Hilos 1-5	/uoc/backusers/add	2240		1860
6	22:22:22.783	Grupo de Hilos 1-6	/uoc/backusers/add	2541		1860
7	22:22:22.884	Grupo de Hilos 1-7	/uoc/backusers/add	2481		1859
8	22:22:22.985	Grupo de Hilos 1-8	/uoc/backusers/add	2843		1860
9	22:22:23.085	Grupo de Hilos 1-9	/uoc/backusers/add	3046		1860
10	22:22:23.188	Grupo de Hilos 1-10	/uoc/backusers/add	3096		1860
11	22:22:23.289	Grupo de Hilos 1-11	/uoc/backusers/add	3444		1860
12	22:22:23.691	Grupo de Hilos 1-15	/uoc/backusers/add	3849		1859
13	22:22:23.389	Grupo de Hilos 1-12	/uoc/backusers/add	4175		1860
14	22:22:24.192	Grupo de Hilos 1-20	/uoc/backusers/add	5061		1859
15	22:22:23.590	Grupo de Hilos 1-14	/uoc/backusers/add	6048		1860
16	22:22:23.490	Grupo de Hilos 1-13	/uoc/backusers/add	6227		1860
17	22:22:24.695	Grupo de Hilos 1-25	/uoc/backusers/add	6381		1859
18	22:22:24.996	Grupo de Hilos 1-28	/uoc/backusers/add	6952		1859
19	22:22:25.397	Grupo de Hilos 1-32	/uoc/backusers/add	7504		1859
20	22:22:25.498	Grupo de Hilos 1-33	/uoc/backusers/add	7655		1859
21	22:22:24.092	Grupo de Hilos 1-19	/uoc/backusers/add	9162		1860
22	22:22:23.791	Grupo de Hilos 1-16	/uoc/backusers/add	9511		1860
23	22:22:23.891	Grupo de Hilos 1-17	/uoc/backusers/add	9412		1860
24	22:22:23.992	Grupo de Hilos 1-18	/uoc/backusers/add	9485		1860
25	22:22:25.899	Grupo de Hilos 1-37	/uoc/backusers/add	8206		1859
26	22:22:26.201	Grupo de Hilos 1-40	/uoc/backusers/add	8484		1859

Extracto del resultado de 20 peticiones.

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes
1	22:20:19.670	Grupo de Hilos 1-1	/uoc/backusers/add	1491		1860
2	22:20:19.720	Grupo de Hilos 1-2	/uoc/backusers/add	1899		1860
3	22:20:19.771	Grupo de Hilos 1-3	/uoc/backusers/add	2215		1860
4	22:20:19.821	Grupo de Hilos 1-4	/uoc/backusers/add	2340		1860
5	22:20:19.871	Grupo de Hilos 1-5	/uoc/backusers/add	2396		1860
6	22:20:19.922	Grupo de Hilos 1-6	/uoc/backusers/add	2497		1860
7	22:20:19.980	Grupo de Hilos 1-7	/uoc/backusers/add	2584		1860
8	22:20:20.030	Grupo de Hilos 1-8	/uoc/backusers/add	2566		1860
9	22:20:20.081	Grupo de Hilos 1-9	/uoc/backusers/add	2582		1860
10	22:20:20.131	Grupo de Hilos 1-10	/uoc/backusers/add	2597		1860
11	22:20:20.182	Grupo de Hilos 1-11	/uoc/backusers/add	4149		1860
12	22:20:21.190	Grupo de Hilos 1-31	/uoc/backusers/add	3288		1859
13	22:20:21.659	Grupo de Hilos 1-40	/uoc/backusers/add	3760		1859
14	22:20:22.015	Grupo de Hilos 1-47	/uoc/backusers/add	4493		1859
15	22:20:22.166	Grupo de Hilos 1-50	/uoc/backusers/add	4758		1859
16	22:20:20.232	Grupo de Hilos 1-12	/uoc/backusers/add	6728		1860
17	22:20:22.267	Grupo de Hilos 1-52	/uoc/backusers/add	5030		1859
18	22:20:20.282	Grupo de Hilos 1-13	/uoc/backusers/add	7035		1860
19	22:20:22.468	Grupo de Hilos 1-56	/uoc/backusers/add	5639		1859
20	22:20:22.568	Grupo de Hilos 1-58	/uoc/backusers/add	5712		1859
21	22:20:22.619	Grupo de Hilos 1-59	/uoc/backusers/add	5956		1859
22	22:20:22.669	Grupo de Hilos 1-60	/uoc/backusers/add	6103		1859
23	22:20:20.434	Grupo de Hilos 1-16	/uoc/backusers/add	9763		1860
24	22:20:20.484	Grupo de Hilos 1-17	/uoc/backusers/add	9723		1860
25	22:20:20.383	Grupo de Hilos 1-15	/uoc/backusers/add	9909		1860
26	22:20:20.333	Grupo de Hilos 1-14	/uoc/backusers/add	10079		1860

Conclusiones de Test Realizados.

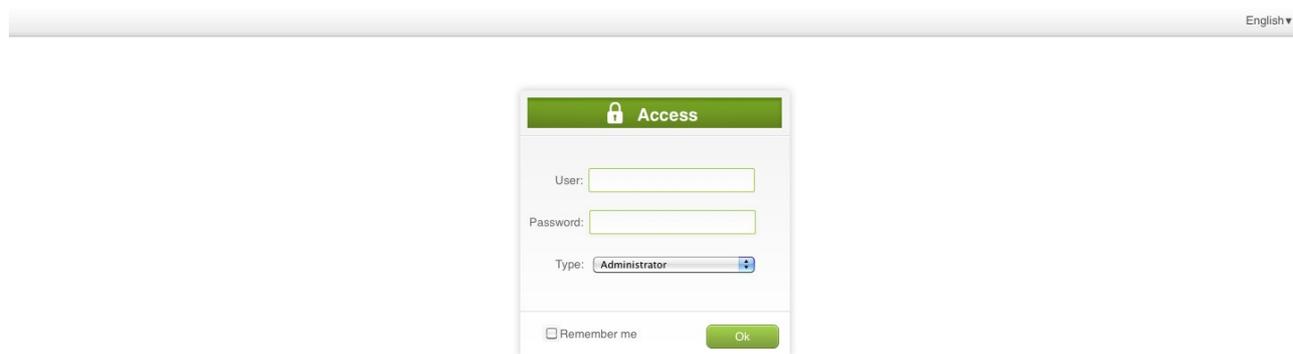
Teniendo en cuenta los resultados mostrados se puede concluir que el servidor de pruebas no está preparado para recibir 10 o 20 peticiones por segundo. Aunque en las primeras 10 peticiones suele responder bien, cuando llegan las siguientes 10 el tiempo de respuesta se incrementa de forma muy notable en cambio cuando se trata de 1 petición por segundo el resultado es muy satisfactorio.

Para finalizar este apartado hay que indicar que aunque el tiempo de respuesta haya sido muy lento en los casos de 10 y 20 peticiones en ninguno el servidor ha dejado de funcionar con lo que se demuestra la estabilidad del hardware y el software.

5.1.7 Algunas pantallas de la aplicación en funcionamiento.

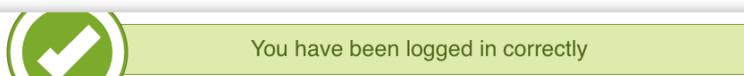
En esta sección se muestran unos pantallazos generales de la aplicación. Podemos ver en el resultado que a nivel de interfaz se ha cumplido con el objetivo.

Pantalla de inicio de sesión.



Pantalla inicial una vez se haya hecho login. Dependerá del usuario que acceda.

Id	Name	Email	Phone	Cellphone	Country	SMS	Product	Agendas	Sign up Date	Expiration Date	
3	First Client	prueba@prueba.com	3333	22222	Spain	10	Bookitit TRIALmonths Standard	10	2012-05-27 17:18:03	2012-06-20 00:00:00	● ✕ ✎ ✖
4	Fourth Client	cuarto@cuarto.com	23423423	23423432423	Spain	20	Bookitit TRIALmonths Standard	10	2012-05-27 17:58:07	2012-06-30 00:00:00	● ✕ ✎ ✖
5	Jmeter	jmeter@jmeter.com	2342343	2343433	Spain	25	Bookitit TRIALmonths Standard	10	2012-05-29 19:52:13	2012-08-01 00:00:00	● ✕ ✎ ✖
6	Request Jmeter	jmet@jmet.com	111111	22222	Spain	20	Bookitit months Standard x 12	10	2012-05-29 20:29:20	2012-08-01 00:00:00	● ✕ ✎ ✖
2	salvador center	bmoliner@prueba.com	605073381	605073381	Spain	100	Bookitit months Multiuser x 12	5	0000-00-00 00:00:00	2014-01-01 00:00:00	● ✕ ✎ ✖
1	Raul Center	raurope@gmail.com		0123456789	Spain	5	Bookitit months Multiuser x 12	5	2010-06-08 05:06:08	2014-01-01 01:00:00	● ✕ ✎ ✖



En esta sección se muestran los datos referentes a los clientes. Además se pueden realizar

diferentes acciones desde ella, como búsquedas activar o desactivar usuarios, eliminar, etc... Esta pantalla es muy similar para el resto de usuarios con la salvedad de que usuarios como partners o colaboradores tendrán que solicitar diferentes acciones que tendrán que ser aceptadas o rechazadas por el administrador.

Sección de Timeline

text	date	Accepted
There is a request to create a new Client. CLIENT NAME: Request Jmeter CLIENT ID: 6	2012-05-29 20:29:20	Accepted
There is a request to create a new Client. CLIENT NAME: Fourth Client CLIENT ID: 4	2012-05-27 17:58:07	Accepted
Your request has been accepted. CLIENT NAME: First Client CLIENT ID: 3	2012-05-27 17:35:05	✘ Accepted
You sent a request to accept or reject an invoice. CLIENT NAME: Raul CLIENT ID: 3	2012-05-27 17:33:55	✘ Accepted
There is a request to create a new Client. CLIENT NAME: First Client CLIENT ID: 3	2012-05-27 17:18:03	Accepted

Desde esta sección se verán listadas las solicitudes que realicen o reciban los usuarios. En la zona de la derecha podremos identificar las contestaciones a cada uno de los mensajes. Cuando se envía una solicitud puede aparecerá un estado aceptar y un estado rechazar para que el usuario decida que hacer con la solicitud.

Sección de Colaboradores

id	company_name	contact_name	email	phone	document	countries_id	regions_id	admprofiles_id	
2	UNIKOFT	antonio	puoc92@yahoo.es		23423432	Spain	Palencia	Partner	✘
3	PARNALIA	nuria	cuoc92@yahoo.es		555555	Spain	Palencia	Colaborador	✘
4	Clientsoft	Clientsoft Clientsoft	cliuoc92@yahoo.es		2342347Y	Spain	Palencia	Client	✘

En este área se pueden gestionar los diferentes colaboradores o partners que participaran en el

sistema. Cada uno de ellos tendrá su propio interfaz.

5.2. Hitos principales del proyecto.

En esta sección hablaremos de los hitos y tareas principales del proyecto. En el proyecto actual se han detectado las siguientes tareas generales:

1. Especificación de Requerimientos y planificación.

En la especificación de requerimientos se definen las bases y requisitos para la construcción del proyecto. (5.1.1, 5.2 y 5.3)

2. Análisis y diseño.

En esta fase se realizará un análisis de las especificaciones construidas en el hito anterior utilizando casos de uso resumidos y detallados. Además se completará la fase de diseño que incluirá bocetos con mockups de los interfaces, diagrama de clases y diagrama entidad relación de base de datos. También se tendrá que añadir a la memoria del plan de trabajo la parte referente a estas dos secciones (5.1.2 y 5.1.3).

3. Implementación de las diferentes partes del software.

Este hito será el que durará más tiempo y consistirá en la construcción del software. Como se ha comentado en los riesgos, durante la ejecución del proyecto podrán haber retrasos, en ese caso se hará especial hincapié en conseguir el desarrollo de la gestión colaborativa que es la más relacionada con el proyecto. Se tendrá que completar la sección correspondiente en el plan de trabajo (5.1.4).

4. Instalación en un sistema uoc/no uoc para demostración.

Se instalará el software para una demostración en un servidor (habilitado por la uoc o por el autor del proyecto). Además se podrá asignar algo de tiempo para finalizar alguna tarea de desarrollo que no se haya terminado.(5.1.5).

5. Evaluación mediante herramientas de test estrés.

Utilización de software de carga de servidores para comprobar como responde el sistema ante situaciones de estrés y completar un resumen(5.1.6) con los resultados obtenidos.

6. Revisión de memoria, valoración económica, entregables y final.

Finalización de la memoria actual, revisión, bibliografía, etc.

Teniendo en cuenta que la tarea 1 se incluye en la PEC1 Plan de trabajo. 26 de Marzo de 2012, de las tareas comentadas anteriormente definiremos 4 hitos:

PEC2_Parcial – 5 de Abril de 2012. Tarea 2.

PEC2_Final - 28 de Abril de 2012. Tareas 2 y parcial de la tarea 3.

PEC3_Parcial – 28 de Mayo de 2012. Finalización tarea 3 y 4.

PEC3_Final - 4 de junio de 2012. Tareas 5 y 6.

5.3. Calendario de trabajo.

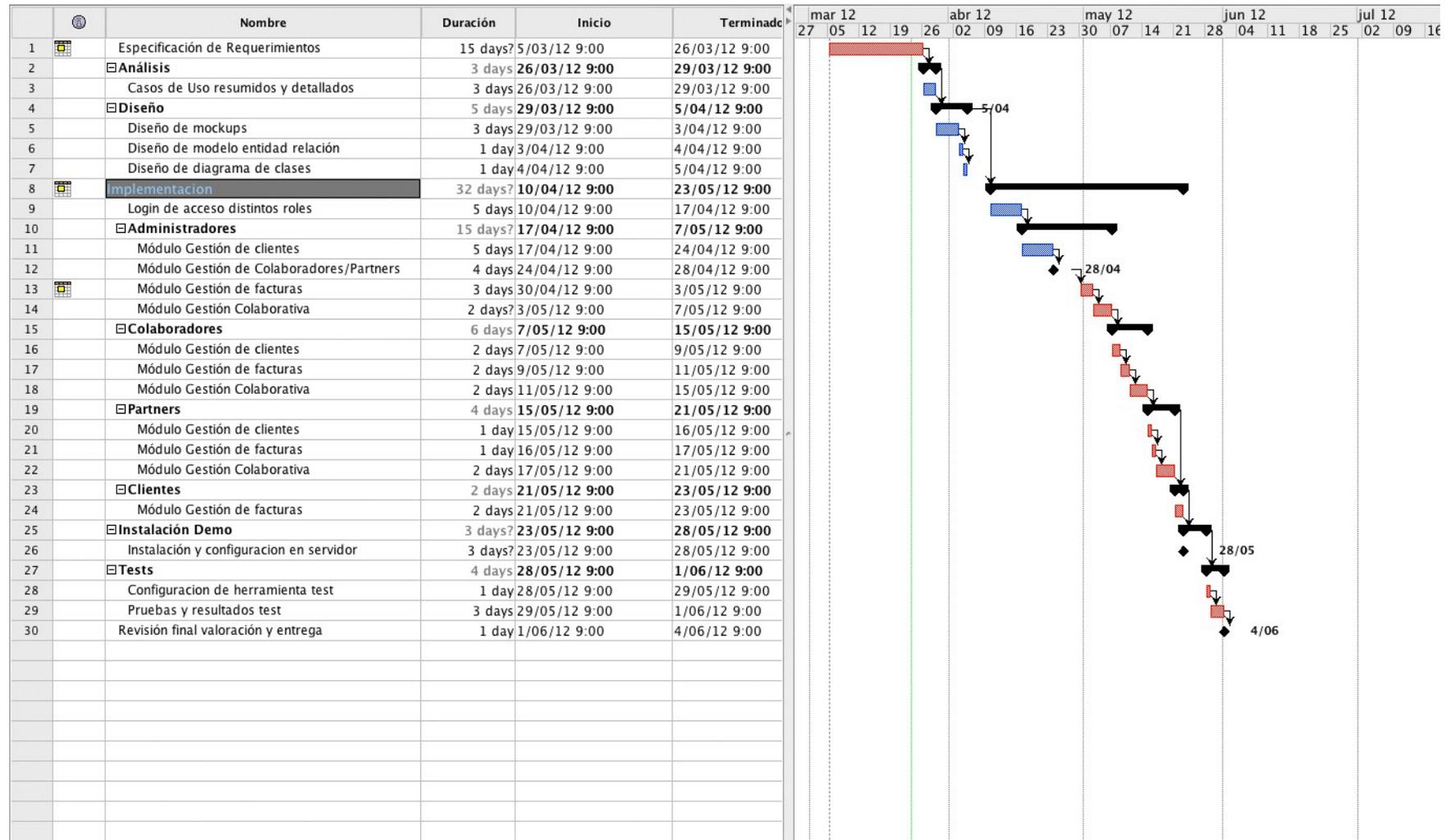
En esta sección se planificará el proyecto desglosado en tareas más concretas y especificando las fechas de entrega. Para la elaboración del gantt se ha utilizado la herramienta Open project.

En el proyecto actual se establecen dos fechas claves: 28 de Abril de 2012 y 4 de junio de 2012. No obstante se establecerán fechas de entrega intermedias para un seguimiento más real del proyecto, teniendo en cuenta las fechas que comentaba anteriormente.

En concreto se establecen 4 fechas de entrega en total. Todas tendrán como tiempo límite de entrega las 23:59:59. A continuación muestro un resumen de las fechas de entrega además de un gráfico gantt con el desglose de las tareas.

Nombre PEC	Descripción	Fecha de entrega
PEC1_Plan_deTrabajo	Esta práctica contiene especificación del proyecto, hitos, fechas de entrega. Hito 1.	26 de Marzo de 2012
PEC2_Parcial	En esta entrega se resolverán las tareas referentes a análisis y diseño. Hito 2	5 de Abril de 2012
PEC2_Final	En esta tarea se iniciará la fase de implementación. Esta fase es la más larga del proyecto por eso se ha dividido en 2 entregas. Para esta PEC2_Final se incluirá análisis y diseño, implementación de login de distintos roles, implementación para administradores gestión de clientes, colaboradores/partners. Hito 3.	28 de Abril de 2012
PEC3_Parcial	En esta PEC se finalizará toda las tareas referente a implementación. Además se realizará la instalación en un servidor para que se puede probar la demostración de la herramienta. Hito 4.	28 de Mayo de 2012
PEC3_Final	Finalización del proyecto, en esta última fase completará la tarea de pruebas de estrés, además de realizar un informe de los resultados y revisar la entrega final de la memoria del proyecto. Hito 5.	4 de junio de 2012

A continuación podemos ver el gráfico Gantt con el desglose de tareas. Podemos observar en el área gráfica las fechas de los distintos hitos y que contendrá cada uno de ellos.



6. Organización del proyecto.

6.1. Equipo de trabajo.

El proyecto actual únicamente dispondrá de un recurso para su desarrollo. La persona encargada tendrá que planificar, diseñar, desarrollar y completar la documentación necesaria para los entregables.

6.2. Definición de roles.

Recurso: Raúl Ros → Responsable del proyecto en todas sus fases.

6.3. Mecanismos de control.

Como mecanismos de control se han establecido unos hitos o puntos de entrega que serán revisados por el consultor. Además para el desarrollo del proyecto se ha utilizado un control de código fuente subversion que permita un correcto control de diferentes versiones de la herramienta.

7. Valoración Económica.

Teniendo en cuenta el calendario establecido y los recursos empleados la valoración económica sería la siguiente:

Fase	Tiempo (en días)
Especificación de requerimientos	15
Análisis	3
Diseño	5
Implementación	32
Instalación Demostración	3
Tests	4
Revisión final valoración y entrega	1

Teniendo en cuenta una media de 8 horas diarias de jornada y suponiendo 40 Euros/hora, la valoración final:

$$63 \text{ días} * 8 * 40 = 20160 \text{ Euros} + \text{IVA}$$

No se ha incluido en la valoración costes por alojamiento o hardware necesario para funcionamiento de la aplicación.

8. Bibliografía.

GANYET M, Josep. UOC. Interacción humana con los Ordenadores.

LAMARCA, Ignacio. UOC. Metodología y gestión de proyectos informáticos.

CABOT, Jordi. UOC. Ingeniería del software orientada a objetos.

CAMPS, Josep. UOC. Ingeniería del software de componentes y sistemas distribuidos.

BENET, Falgueras. Desarrollo del software orientado al objeto.

EGUILIZ, Javier. CSS Avanzado.

EGUILIZ, Javier. Introducción a Javascript.

GAUCHAT, J.D. El gran libro de HTML5, CSS3 y Javascript.

ZANDSTRA, MAT. Php Objects, patterns and Practice.

Otros recursos web utilizados.

<http://biscminds.blogspot.com.es/2011/12/quick-jmeter-setup-on-mac.html>

<http://jmeter.apache.org/>

http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

<http://www.webestilo.com/php/>