Concepción de un sitio web de soporte técnico para los proyectos SIDUNEA World

Proyecto fin de máster Software Libre

Administración Web y Comercio Electrónico

Laurent Rey

Consultor UOC : Francisco Javier Noguera Otero
Tutor externo : Yannick Goujon

Lusaka, Zambia, 12 de enero 2014

Usted es libre de :

Compartir - copiar, distribuir, ejecutar y comunicar públicamente la obra

Re-mezclar – transformar la obra y hacer obras derivadas

Bajo las condiciones siguientes:

Atribución — Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciante (pero no de una manera que sugiera que tiene su apoyo o que apoyan el uso que hace de su obra).

No Comercial — No puede utilizar esta obra para fines comerciales.

Compartir bajo la Misma Licencia — Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

Entendiendo que:

Renuncia — Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor

Dominio Público — Cuando la obra o alguno de sus elementos se halle en el dominio público según la ley vigente aplicable, esta situación no quedará afectada por la licencia.

Otros derechos — Los derechos siguientes no quedan afectados por la licencia de ninguna manera:

- Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.
- Los derechos morales del autor;
- Derechos que pueden ostentar otras personas sobre la propia obra o su uso, como por ejemplo derechos de imagen o de privacidad.

Aviso — Al reutilizar o distribuir la obra, tiene que dejar muy en claro los términos de la licencia de esta obra.

Resumen

El presente proyecto consiste en el desarrollo y la implantación en producción de un sitio de web de soporte técnico para los proyectos SIDUNEA World de la Conferencia de la Naciones Unidas sobre Comercio y Desarrollo.

SIDUNEA es un sistema de gestión de las operaciones aduaneras automatizado que cubre la mayor parte de trámites de comercio exterior. El sistema propone una solución integrada para el proceso de manifiestos, declaraciones de aduana, procedimientos de contabilidad, tránsito y regímenes suspensivos. SIDUNEA puede configurarse y adaptarse a las distintas necesidades nacionales de cada administración aduanera, aranceles nacionales, leyes y reglamentación aduanera.

Este sitio web sera compuesto de dos partes. La primera parte sera estática que suministrará informaciones generales sobre el programa SIDUNEA World y sus implantaciones. La segunda parte sera dinámica en el sentido que sera el vector principal para suministrar y compartir informaciones técnicas entre el equipo central de desarrollo basado en Ginebra y todos los actores involucrados en los proyectos SIDUNEA World establecidos en los países. Este proyecto se apoya sobre los software libres y sigue los estándares del World Wide Web Consortium (W3C).

Índice general

1 INTRODUCCIÓN....5
2 PLANIFICACIÓN....7
2.1 Planificación del sistema de información....7
2.1.1 Estudio de viabilidad....7
2.1.2 Identificación de los actores y usuarios....8
2.1.3 Definición de los requisitos del sistema....9
2.1.4 Estudio de las alternativas de solución....10
2.1.5 Alcance del sistema....12
2.1.6 Análisis de riesgos....12
2.1.7 Diagrama EDT....14
2.2 Análisis....15
2.2.1 Definición del sistema y Requisitos....16
2.2.2 Interfaces de usuarios....18
2.2.3 Plan de prueba....24
2.3 Diseño....25
2.3.1 Arquitectura del sistema....25
2.3.2 Elección de software....29
2.3.3 Requisitos de implantación....30
3 EJECUCIÓN....31
3.1 Desarrollo 31
3.1.1 Planificación de las actividades....31
3.1.2 Desarrollo del proyecto....32
3.2 Implantación....51
3.2.1 Instalación en entorno de producción....51
3.2.2 Pruebas de implantación....52
3.2.3 Nivel de servicios....53
4 CONCLUSIONES....54
5 BIBLIOGRAFÍA....55

1 INTRODUCCIÓN

El presente documento ha sido redactado en el marco del trabajo final de aplicación profesional del Máster Universitario en Software Libre para poner a disposición

- del consultor de UOC encargado de la asignatura Proyecto Final de Máster de orientación profesional
- del tutor externo en su calidad de director de proyecto
- de los miembros del tribunal de proyecto
- de los lectores

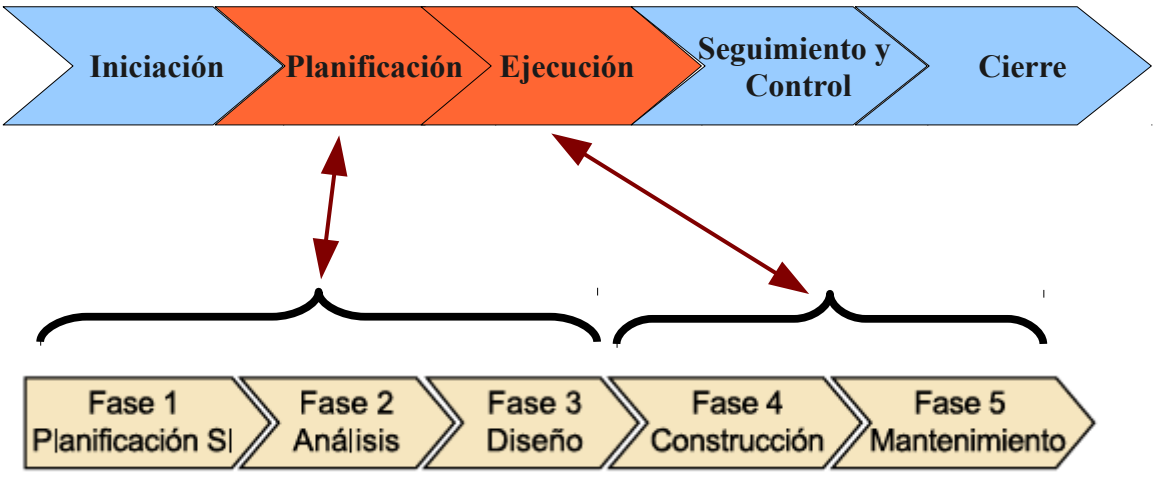
las informaciones más relevantes con respecto al proyecto de desarrollo del sitio web de soporte técnico para el programa de SIDUNEA World de la Conferencia de la Naciones Unidas sobre Comercio y Desarrollo.

Este proyecto se apoya sobre el ciclo de vida de desarrollo de sistemas (SDLC - System Development Life Cycle) que se compone de 5 fases: Planificación SI, Análisis, Diseño, Construcción y Mantenimiento.

Las fases del SDLC hacen parte del PMBOK (Project Management Body of Knowledge) que es un conjunto de conocimientos en Dirección/Gestión/Administración de Proyectos generalmente reconocidos como "buenas prácticas", y que se constituye como estándar de Administración de proyectos.

El PMBOK se compone de cinco grupos básicos de procesos donde cada uno se divide en áreas de conocimientos.

Fases de PMBOK



Fases de SDLC

Siguiendo las fases del SDLC, los capítulos de este documento son los siguientes:

1. **Introducción** : el capitulo de introducción presenta el objetivo y la descripción general del contenido de este documento.
2. **Planificación** : el capítulo de planificación contiene los parágrafos siguientes :
    - **Planificación del sistema de información** : este parágrafo definiremos los actores interno y externo del proyecto, haremos la recopilación de los requisitos que nos permitirá definir el alcance del proyecto que es una primera estimación de los objetivos y resultados esperados. La identificación de los riesgos ayudará a prever las respuestas adecuadas y por fin el diagrama EDT (Estructura de Descomposición del Trabajo) nos dará una vista de los entregables del proyecto.
    - **Análisis** : en el parágrafo del análisis profundizaremos los requisitos, especificaremos las interfaces de los usuarios y sus interacciones con el sistema y por fin definiremos el plan de prueba.
    - **Diseño :** en el parágrafo de diseño definiremos las arquitectura del sistema así como los estándares que seguiremos para el desarrollo. Terminaremos este parágrafo con los requisitos de implantación.
3. **Ejecución** : el capitulo de ejecución contiene los parágrafos siguientes :
    - **Desarrollo** : en el parágrafo de desarrollo identificaremos las actividades y tareas a hacer para proponer el plan de trabajo y el calendario de hitos. Describiremos también las etapas y puntos relevantes de cada subsitema desarrollado del sitio web y sus pruebas de validación.
    - **Implantación** : en el parágrafo de implantación describiremos las etapas de implantación en producción del sitio web y las pruebas de validación de instalación. Sera el momento de presentar el nivel de servicio del sistema.
4. **Conclusiones** : el capitulo de las conclusiones contiene los resultados alcanzados y sugerencias de mejoras futuras. Será también el momento del análisis al nivel personal de esta experiencia, de resumir las competencias adquiridas, los éxitos y dificultades.
5. **Anexos** : en el capitulo de los anexos, buscaremos toda la documentación relevante que nos permitirá obtener las informaciones técnica y funcional del proyecto. (manuales de usuario, descripción del modelo de la base de datos, configuraciones...)

Esta memoria presenta el estudio del desarrollo de un sitio web desarrollado con software libre. No tratará de las reglas sintácticas y semánticas de los lenguajes de programación que utilizaremos, tampoco de las instalaciones de los programas (herramientas, base de datos, servidor web …) que pondremos en marcha para nuestro sistema. Por supuesto aprovecharemos para detallar ciertas funcionalidades relevantes.

Este documento pone en aplicación una parte de los conocimientos adquiridos durante el Máster en Software libre y se presenta de manera que no sea necesario tener conocimientos técnicos avanzados.

2 PLANIFICACIÓN

2.1 Planificación del sistema de información

En este capitulo se presenta los aspectos del estudio de viabilidad del proyecto, cubriendo los parágrafos siguientes :

- **Estudio de viabilidad :** en el estudio de viabilidad se presenta la situación actual y las primicias del proyecto.
- **Identificación de los actores e usuarios :** en este parágrafo se establece la lista de los usuarios e actores impactados por el proyecto.
- **Definición de los requisitos del sistema :** según los parágrafos previos se presenta los requisitos generales del proyecto.
- **Alcance del sistema :** es la descripción de las necesidades planteadas por el cliente.
- **Análisis de riesgo :** Identificación de los riesgos y las respuestas adecuadas para paliar los efectos negativos de los riesgos..
- **Diagrama EDT :** nos dará una vista exhaustiva de los entregales del proyectos.

2.1.1 Estudio de viabilidad

Desde 1981, varias versiones de SIDUNEA fueron desarrolladas para modernizar los procedimientos de desaduanamiento y producir estadísticas fiables del comercio exterior de los países que implantaron el proyecto. Estas versiones siempre fueron “cerradas”. Los países que instalaban SIDUNENA no tenían la posibilidad de modificar, adaptar y en ciertos casos corregir la versión instalada. El equipo central de Ginebra se encargaba del desarrollo y suministraba con frecuencia actualizaciones integrando los requerimientos comunes.

Gracias a la ultima versión de la plataforma del SIDUNEA, SIDUNEA World que se compone de documentos electrónicos, los países aprovechan de una versión estándar del sistema de gestión aduanero SIDUENA y ahora tienen la posibilidad para adaptar el programa a su reglamentación aduanera y código aduanero, desarrollar y integrar documentos utilizados por la administración aduanera local bajo formato electrónico. Eso se puede hacer debido a que SIDUENA World se distribuye con su código abierto y los expertos de la UNCTAD tienen como misión de capacitar los equipos locales para la administración el sistema pero también para el desarrollo de documentos electrónicos.

Con las previas versiones de SIDUNEA, el equipo central de Ginebra recibía

únicamente demandas de soporte para la parte funcional pero ahora las peticiones para el soporte técnico aumentaron de manera drásticas. La respuesta a estas peticiones por el correo electrónico ahora no es suficiente y eficiente.

Una solución que incluye a todos los actores de los proyectos es urgente debido a un flujo de demanda exponencial. Además las respuestas interesan a todos y deben ser compartidas.

Cada proyecto tiene sus propios problemas pero estos pueden acercarse de los demás y compartir, intercambiar ideas permite mejorar cada unos de los proyectos.

Actualmente, existe un sitio web del programa SIDUNEA, www.asycuda.org pero son solamente noticias sobre los nuevos convenios o acuerdos de implementación del programa, no existe un espacio que trata de los aspectos técnicos que toman hoy en día una parte muy importante.

Se ha detectado la necesidad de un sitio web de soporte técnico para las razones siguientes:

- Una nueva manera de comunicar por parte del proyecto SIDUNEA World.
- Crear una comunidad que intercambia informaciones técnicas como objetivo de mejorar el sistema.
- Consolidar el sistema con las informaciones devueltas por partes de todos los actores del proyecto.
- Difundir con eficacia y rapidez los parches, correcciones de errores, ejemplos de código.
- Informar sobre las nuevas versiones, nuevos módulos desarrollados.
- Poner a disposición para los usuarios la documentación técnica y funcional, las versiones del proyecto y herramientas.

Al nivel de los equipos locales de cada país y consultores los beneficios serán :

- de traer una dinámica al nivel de los proyectos;
- de dar la oportunidad a los proyectos locales de presentar sus desarrollos y evoluciones que integraron en SIDUNEA World a fin de que los demás puedan conocerlas y estudiar la posibilidad de implementarlas;
- de favorecer el intercambio de ideas y innovaciones;
- de crear canales de comunicaciones entre los proyectos.

2.1.2 Identificación de los actores y usuarios

Podemos identificar 3 grupos distintos de usuario del sito web.

El primero grupo se compone de cualquiera que busca información sobre el programa SIDUNEA World, sus ventajas y sus implantaciones. Para satisfacer su curiosidad tendrá a su disposición la parte estática del sitio web que le dará todas las informaciones sobre las posibilidades ofrecidas de este programa.

El segundo grupo se compone de los usuarios establecidos en los países que implementan el programa SIDUNEA World. Estos usuarios tendrán acceso a la parte

dinámica del sitio web mediante una cuenta con login/contraseña. Los diferentes actores de este segundo grupo son:

- Los expertos UNCTAD, consultores, desarrolladores de los proyectos, los miembros del equipo nacional de proyecto se dividen en 2 tipos de usuario:
  - los usuarios que tendrán acceso a las informaciones suministradas por parte de Ginebra, principalmente las noticias técnicas, documentaciones, versiones del programa, ejemplos de códigos, consejos sobre buenas practicas. Cualquier usuario que tendrá una cuenta de login.
  - Los usuarios que podrán actualizar las informaciones de la pagina de su proyecto SIDUNEA World. Podrán difundir descripciones sobre nueva implementación de documento electrónico, integración y desarrollo de nuevas funcionalidades en los módulos de la versión estándar de SIDUNEA World, suministrar SPR (System Problem Report)

El tercero grupo de usuarios se compone principalmente de los miembros del equipo central de Ginebra encargado del programa SIDUNEA World.

- Los miembros del equipos central de Ginebra se dividen en 2 tipos de usuarios:
  - Responsables del programa y los desarrolladores podrán redactar las noticias y las informaciones técnicas, difundir versiones y documentaciones, manejar el sistema de gestión de SPR (System Problem Report).
  - Los usuarios con derechos de administración para administrar las cuentas de los usuarios y mantener los datos del sistema.

2.1.3 Definición de los requisitos del sistema

Los requisitos principales del sitio web son los siguientes :

Requisitos técnicos:

- El proveedor del alojamiento del sitio web debe suministrar una interfaz para administrar el sistema y su instalación mediante la utilización de cualquier navegador.
- El proveedor del alojamiento del sitio web debe suministrar las herramientas y accesos para realizar copias de seguridad de los datos y del sitio web.
- Los programas, herramientas, librarías y lenguajes de programación deben ser debajo licencias abiertas.
- La concepción del sitio debe cumplir con los estándares del World Wide Web Consortium (W3C).
- Las implementaciones de las peticiones con la base de datos debe cumplir con el estándar del la norma SQL92.
- La arquitectura de software del sitio web debe seguir el patrón Modelo Vista Controlador (MVC) separando los datos y la lógica de negocio de la interfaz de

usuario.

- El diseño de las interfaces del sitio web debe utilizar las hojas de estilo en cascada versión 3 (CSS3).

Requisitos operativos

- La navegación en el sitio web debe ser intuitiva y agradable.
- Las diferentes secciones y subsecciones deben ser accesibles mediante un menú.
- La parte estática del sitio web sera disponible en 3 idiomas : ingles, francés y español.
- El sitio debe permitir de subir documentos y archivos mediante una interfaz sencilla.
- La gestión del contenido debe ser realizada mediante interfaz web accesible desde un menú especial para los administradores.
- El acceso a la parte dinámica se realizará mediante una cuenta usuario y contraseña .
- Un flujo RSS permitirá de seguir las actualizaciones del sitio web.

Requisitos económicos

- Todos los programas, herramientas, librarías y lenguajes de programación no deben tener costo para la UNCTAD.
- El alojamiento del sitio web debe tener un costo lo más pequeño posible.

2.1.4 Estudio de las alternativas de solución

Nuestro suministrador de alojamiento Inmotion Hosting permite de desarrollar sitios web basado sobre 3 elementos : el interprete PHP, la base de datos MySQL y el servidor web Apache. Para llevar a cabo el desarrollo, Inmotion Hosting da una amplia oferta de framework o CMS (Content Management System) a instalar.

La cuestión es que debemos elegir un CMS o un framwork?

Los CMS son herramientas / programas que ofrecen funcionalidades llave en mano fácil a instalar. Permiten desarrollar un sitio rápidamente y a menor coste. Se necesita pocos esfuerzos para poner en marcha y son accesibles sin necesitar conocimientos técnicos de alto nivel. Sin embargo, sus ventajas hacen también sus inconvenientes. El hecho de tener funcionalidades ya desarrolladas los hacen difícilmente modificables y con menos flexibilidad.

Una aplicación web que contiene llamada a una base datos, formularios, sesiones, conexiones a servicios, aumentara su rendimiento si utiliza un framework.

- Características de un framework

  - **Capa de abstracción de la base de datos** : un framework suministra una capa de abstracción o interfaz entre el código de nuestra aplicaciones y las peticiones hacia el servidor de base de datos. Nos da la posibilidad de conectarnos a varias base de datos desarrollar de nuevo código.
  - **Capa de abstracción del cache** : en lugar de llamar las funciones manejando el cache, se puede utilizar clase genérica encargada de manejar el cache.
  - **Gestión de los formularios** : los frameworks suministran clases para facilitar la implementación de los formularios y la validación de los datos.
  - **Autenticación** : pueden suministrar modulo de autenticación manejando la conexión/desconexion, la gestión de las sesiones...
  - **Internacionalización** : permiten de crear facilamente versiones del sitio o aplicación de acuerdo a las idiomas elegidas.
- Ventajas de un framework :
  - **Portabilidad** : gracias a su nivel de abstracción se puede instalar su aplicación sobre varios configuraciones de servidores diferentes.
  - **Tiempo de desarrollo más corto** : sin tener la obligación de desarrollar de cero una aplicación debido a las clases y herramientas suministrada por el framework.
  - **Seguridad de las aplicaciones** : las funciones de seguridad, sesiones, inserciones en la bases de datos controladas son funciones de los frameworks.
  - **Una comunidad activa** : la mayor parte de los frameworks son con código abierto y es fácil de buscar respuesta a un problema o correcciones.
  - **Plugins y módulos** : los miembros de la comunidad desarrollan plugins y módulos que podemos descargar y aprovechar para aumentar las funcionalidades de nuestro sitio.
  - **Reglas de codificación** : la mayor parte de los frameworks nos obligan a seguir reglas de codificación estricta para lograr a desarrollar aplicaciones respetando la arquitectura del framework. MVC es un modelo de codificación muy utilizado por los framework para construir sitios y aplicaciones web.

La oferta es amplia : Zend, Symfony2, CakePHP, Kohana, Codeigniter, Prado, DIY.

A menudo la elección de un framework es una cuestión de gusto. Después haber leído varias descripciones, elegí Codeigniter porque :

- es un framework ligero,
- necesita sólo poca configuración,
- no impone reglas estrictas de codificación,
- fácil a aprender.

2.1.5 Alcance del sistema

El sitio web de soporte técnico debe incluir como mínimo :

- Una parte estática accesible para el publico que busca informaciones sobre el programa SIDUNEA World y sus implantaciones.
  - Presentación del Centro Regional / Sitio Soporte técnico
  - Presentación de la plataforma SIDUNEA World
  - Presentación de los países de la región que utilizan la plataforma SIDUNEA World : Datos técnicos, sitio de implantación, versión, etc...
  - Página de contacto
  - Página de enlaces : UNCTAD (Conferencia de las Naciones Unidas sobre Comercio y Desarrollo), SIDUNEA, OMC (Organización Mundial del Comercio), OMA (Organización Mundial de Aduanas)...
- Una parte privada dinámica accesible mediante una cuenta de usuario con contraseña. Esta zona privada tiene como funcionalidades:
  - Difundir Noticias (news o blog) proporcionadas por el equipo central del proyecto de Ginebra : noticias técnicas, corrección de bug, modificaciones, nuevos códigos fuentes, informaciones sobre las nuevas funcionalidades, documentaciones funcional y técnica.
  - Noticias de los proyectos : los países podrán difundir sus modificaciones y nuevas funcionalidades desarrolladas mediante un formulario, a fin de que los demás puedan conocerlas y estudiar la posibilidad de implementarlas.
  - Seguimiento de las SPR (System Problem Report) : creación automatizada del reporte SPR, permitiendo notificaciones y seguimiento de los errores encontrados por los países que usan el SIDUNEA World.
  - Información y descarga de las versiones y su respectiva documentación.
  - Acceso al foro de soporte técnico y funcional.
- Dentro la zona privada, un acceso para los administradores del sitio, que tendrá como funcionalidades:
  - Gestión de los usuarios : creación de cuentas, afectación de roles.
  - Gestión de las tablas de la base de datos utilizadas en las combo box de los formularios (lista de los países, versiones del SIDUNEA, lista de los módulos del SIDUNEA, versiones de java, tipo de base de datos, idiomas, etc) .
  - Formulario para crear nuevas noticias (news o blogs)
  - Formulario para poner en línea nuevas versiones y archivos.

2.1.6 Análisis de riesgos

Varios tipos de riegos pueden surgir durante el proyecto. Debemos tomar en serio las posibilidades de evento imponderables. más abajo buscarán la descripción de los diferentes riesgos del proyecto.

Riesgos técnicos:

Riesgos	Prevención	Para el proyecto
Utilización de nuevos lenguajes informático o de una nueva tecnología.	Evaluación de las necesidades de formación del equipo.	Auto-formación y utilización de lenguajes y tecnologías ya utilizados o estudiados.
Deficiencia de la maquina de desarrollo.	Prever respaldo cotidiano del entorno y fuentes del proyecto.	Backup automatizado de la maquina de desarrollo y utilización de un sistema de control de versiones
El proveedor no abastece más unos de los programas que utilizamos.	Recopilar los proveedores que suministran una oferta similar.	Tener una lista de otros proveedores potenciales.
Deficiencia de servicio del proveedor.	Trabajar con el servicio jurídico para el estudio de las protecciones ofrecidas por el proveedor.	Backup automatizado de los datos del sistema y transferencia hacia la oficina como respaldo.

Riesgos humanos:

Riesgos	Prevención	Para el proyecto
Enfermedad, defección de un recurso del proyecto, re-afectación sobre otras prioridades.	Prever a personas “backup” que tendrían las mismas competencias y podrían reemplazarlos en caso de ausencia prolongada.	Dado que soy el único recurso podemos decir que es a la vez un inconveniente y un aventaje.
Re-afectación de recursos sobre otras prioridades operacionales.		
Incompetencia de los recursos con relación a las tareas que se les han sido afectadas.	Selección de los recursos con competencias y experiencias de acuerdo con las exigencias del proyecto.	

Riesgos jurídicos:

Riesgos	Prevención	Para el proyecto
Quiebra del proveedor de	Trabajar con el servicio	Tener una lista de otros

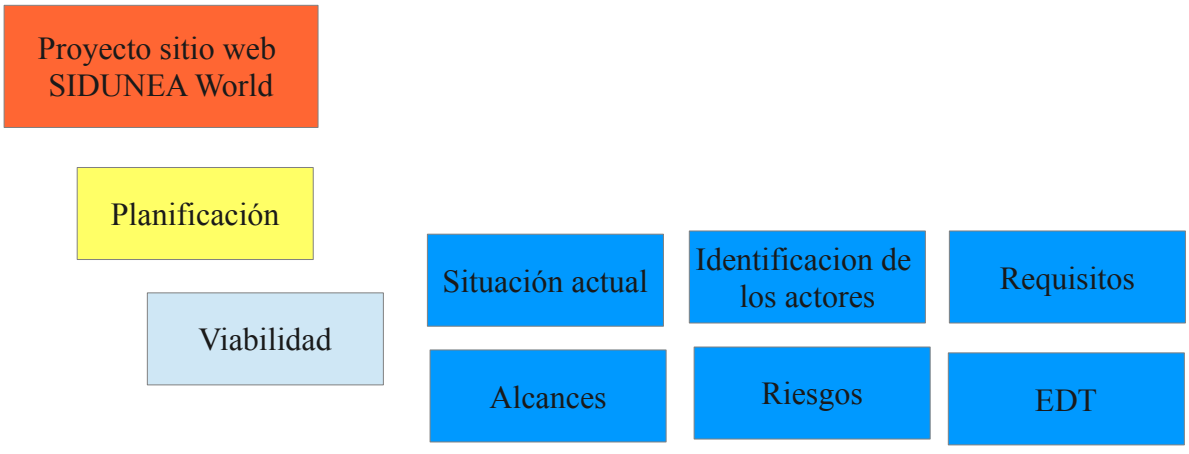
alojamiento del servidor web y de los datos.	jurídico para el estudio de los contratos.	proveedores potenciales.
Equivocación sobre la licencia de un programa o de un extracto de código utilizado.	Seleccionar únicamente programas y códigos debajo licencia libre.	En caso de duda pedir al proveedor las características de su licencia.

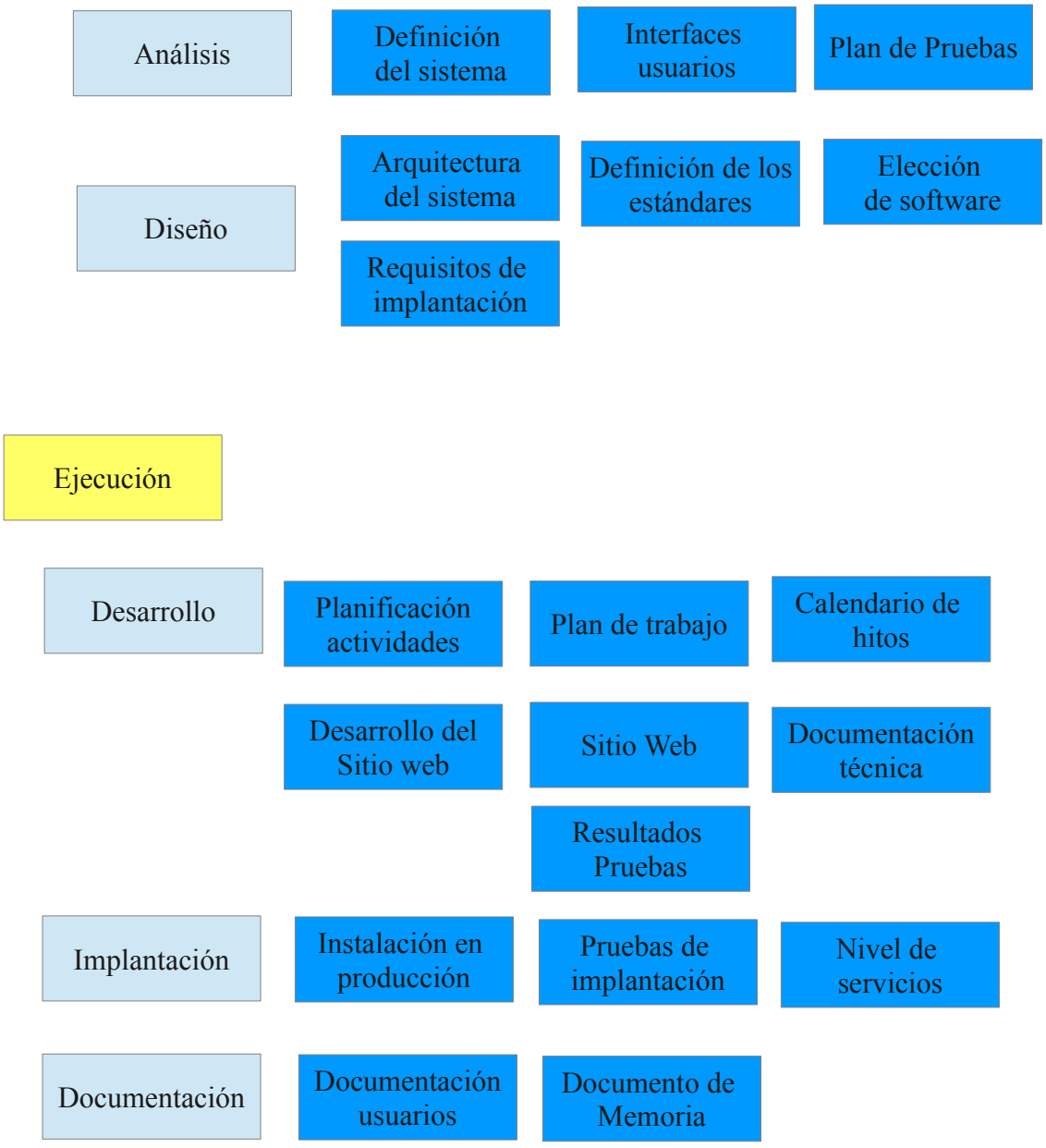
Riesgos intrínsecos a la gestión de proyecto:

Riesgos	Prevención	Para el proyecto
Mala estimación inicial de la duración necesaria para la ejecución las tareas y mala afectación de las responsabilidades sobre las tareas,	Para estimar las cargas, utilizar métricas, o utilizar planificaciones anteriores sobre otros proyectos. Revalidar con los participantes la duración necesaria guardando desde luego una mirada crítica.	Responsable de la planificación y de la estimación de la ejecución de las tareas, tengo una flexibilidad superior a un equipo de varias personas. Sin embargo no se debe descuidar el aspecto de planificación y respetar los hitos.
Mala implicación de las partes interesadas y en particular de uno o varios socios comanditarios del proyecto.	Definir de nuevo la responsabilidades de cada unos.	

2.1.7 Diagrama EDT

El diagrama de Estructura de Descomposición del Trabajo (EDT) nos presenta los entregables del proyecto.





2.2 Análisis

En este capitulo se presenta los aspectos del estudio de análisis del proyecto, cubriendo los parágrafos siguientes :

En el parágrafo del análisis profundizaremos los requisitos, especificaremos las interfaces de los usuarios y sus interacciones con el sistema y por fin definiremos el plan de prueba

- **Definición del sistema y requisitos :** la definición del sistema nos permitirá profundizar en los requisitos presentados en el estudio de viabilidad y llevar a

cabo una especificación detallada de ésta .

- **Interfaces de usuarios :** en este parágrafo se establecerá, teniendo en cuenta los diferentes perfiles de los usuarios, las diferentes interfaces disponibles.
- **Plan de prueba :** El plan de prueba definirá las diferentes pruebas que hay que realizar para establecer si el sistema cumple con los requisitos establecidos.

2.2.1 Definición del sistema y Requisitos

Requisitos de la parte pública:

- acceso a la parte privada por login,
- navegación fácil por menú

Requisitos de la parte privada :

Conexión a la parte privada :

- solamente los usuarios con una cuenta y una contraseña podrán acceder a la parte privada.
- cada página de la parte pública tendrá la opción para conectarse mediante un cuadro con los campos necesarios para llenar su nombre y su contraseña.
- Los errores de cuenta y/o de contraseña serán fijadas en el cuadro que permite concertarse.
- Una vez conectado, cada usuario podrá modificar sus datos mediante una opción del menú Admin denominada My Profile.

Las noticias ;

- Cada usuario conectado podrá :
  - visualizar las noticias,
  - acceder al a dirección del correo electrónico del autor de la noticias para contactarlo,
  - acceder a los comentarios,
  - comentar las noticias,
  - acceder al a dirección del correo electrónico del autor del comentario para contactarlo.

Informaciones sobre los proyectos:

- Cada usuario conectado podrá:
  - consultar los datos de los proyectos por país : informaciones generales, datos técnicos, implantaciones de oficinas de aduana con Asycuda World en una mapa, lista de los diferentes contratos firmados con la UNCTAD y lista de las nuevas funcionalidades / mejoras desarrolladas por el país,
  - visualizar los datos de cada oficina de aduana mediante una mapa generada debajo google maps,
  - cada oficina de aduana podrá ser visualizada en la mapa por un icono. Un

clic en el icono nos dará las informaciones de la oficina seleccionada,

- Cada nueva funcionalidad / mejora sera acompañada de un archivo descargable describiendo en detalle esta mejora.

Descargas :

- Cada usuario conectado podrá:
  - posibilidades de descargar versiones de Asycuda World y/o programas,
  - Posibilidades de descargar documentación funcional o tetánica.

Cada usuario tendrá acceso a los datos de su perfil.

Administración :

De acuerdo al rol afectado al usuario, podrá acceder a diferentes partes de la administración del sistema.

Usuario con el perfil Administrador Noticias, podrá :

- añadir nuevas noticias,
- modificar noticias,
- borrar noticias,
- definir si la noticia debe ser publicada o no,
- añadir la noticia al flujo RSS si esta definida como publicada.

Usuario con el perfil Administrador Descargas, podrá :

- añadir una nueva descarga que sea una version de SIDUNEA World o una documentación,
- modificar las informaciones de la descarga,
- modificar unicamente el archivo descargable,
- borrar descargas,
- definir si la descarga debe ser publicada o no,
- añadir la descargas al flujo RSS si esta definida como publicada.

Usuario con el perfil Administrador Proyecto, podrá :

- manejar las informaciones del proyecto de acuerdo al país definido en su perfil,
- añadir, modificar informaciones generales y técnicas del proyecto,
- añadir, modificar y borrar informaciones sobre las oficinas de aduana, los contratos y las nuevas funcionalidades / mejoras,
- añadir y cambiar el archivo describiendo los detalles del la nueva funcionalidad / mejora,
- para cada nueva información o modificación de un proyecto un flujo RSS será creado automáticamente.

Usuario con el perfil de Administrador :

- tendrá las opciones de todos los roles,

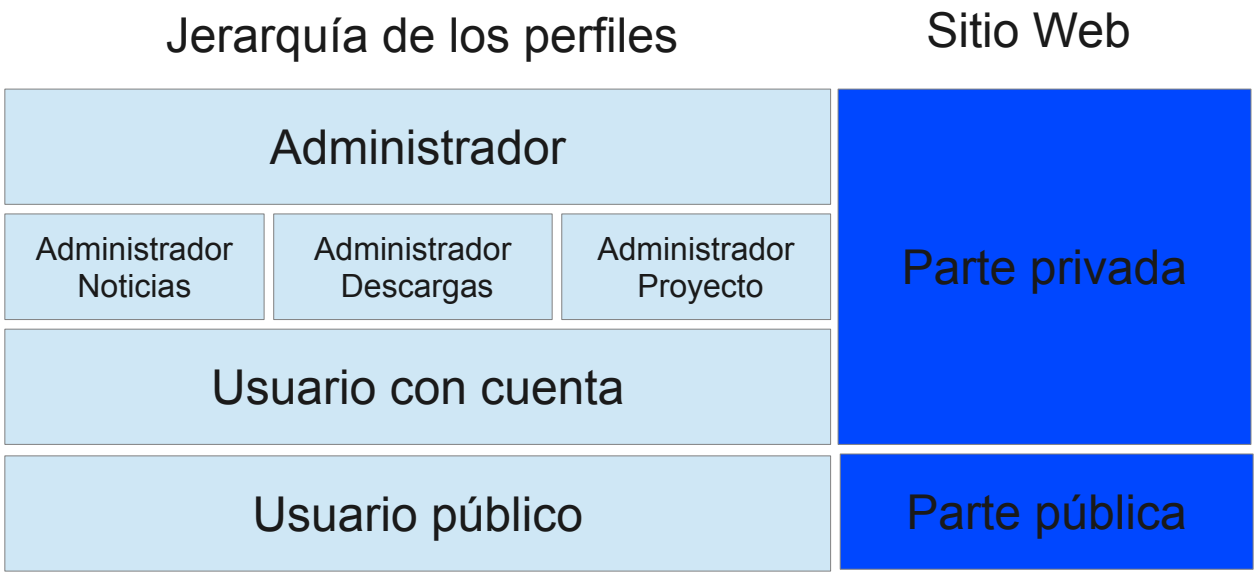
además podrá :

- manejar las cuentas de los usuarios : crear, modificar y borrar,
- manejar las informaciones de las tablas de referencia : añadir, modificar y borrar. (tablas de referencia : zona de proyecto, lista de los países, versión de base de datos, versión de java, versión de SOClass, lista de sistemas operativo, versión de Asycuda World),
- manejar los flujos RSS : crear, modificar y borrar,
- añadir un país a una zona de proyecto,
- borrar un proyecto completo.

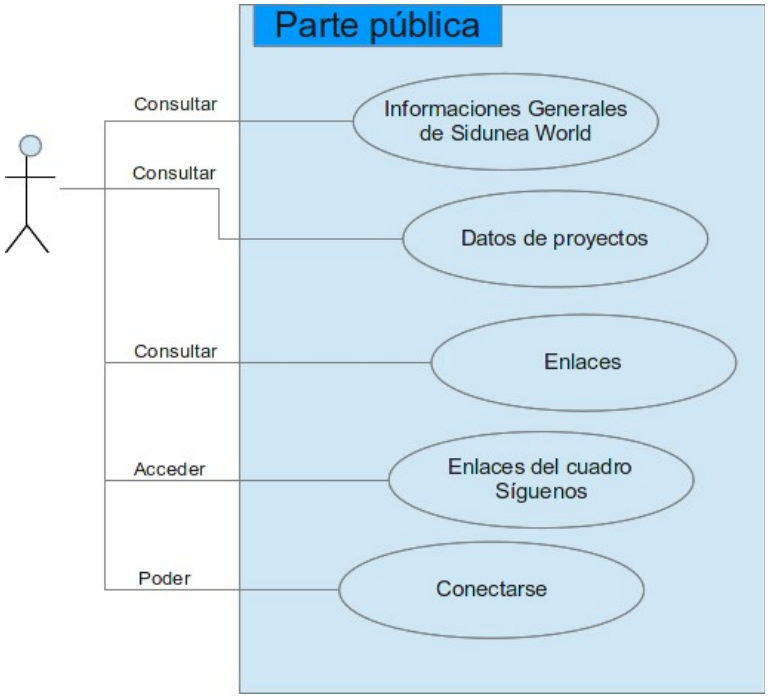
2.2.2 Interfaces de usuarios

Perfiles de usuarios

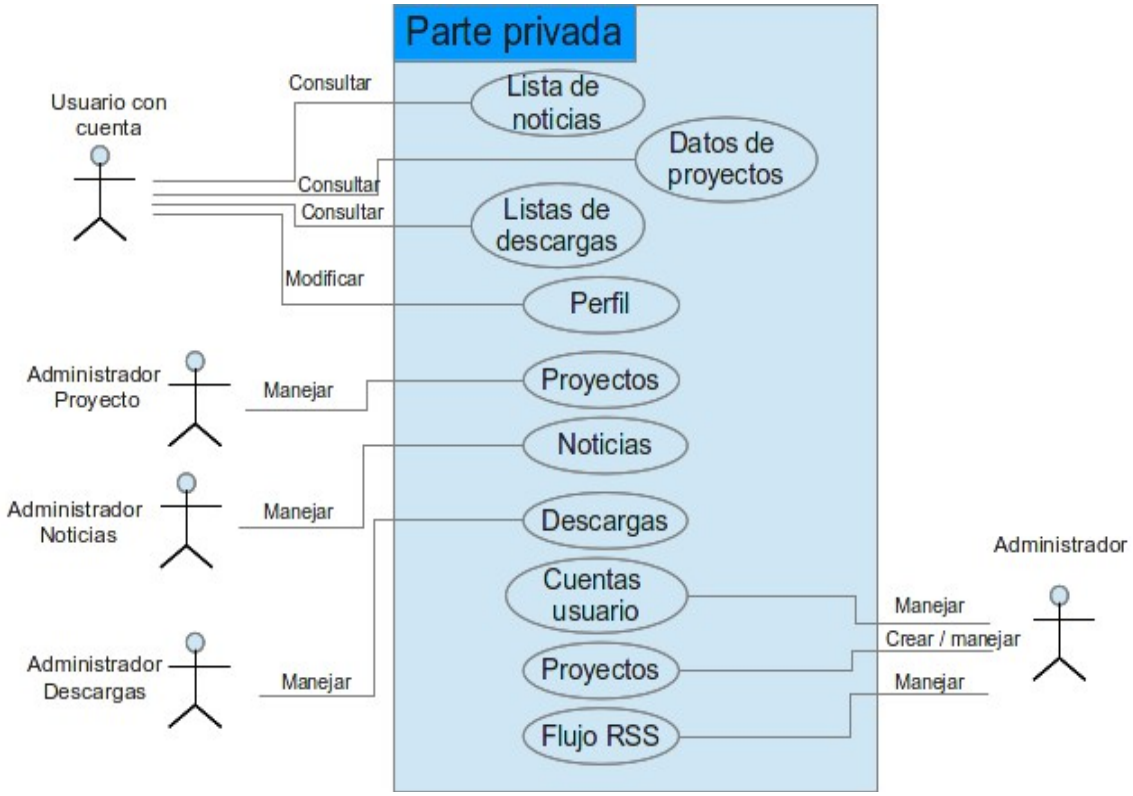
Antes de definir las interfaces se debe hablar de los perfiles de usuarios. En el apartado anterior *2.1.2 Definición del sistema y requisitos*, tenemos la descripción de los diferentes permisos de cada perfil de usuario. Cada usuario accede a las informaciones o a la administración del sitio web de acuerdo a los derechos otorgado por su perfil. Una jerarquía de los perfiles establece que hay un herencia de los derechos cada vez que subimos de un nivel. Se puede acumular derecho de administración, es decir un usuario puede encargarse de la administración de las noticias y las descargas.



Esquema de acceso por la parte pública



Esquema de acceso por la parte privada



Interfaces

El diseño de las interfaces de un sitio web es una parte importante si no la más importante debido a que es lo que los usuarios verán y utilizaran para buscar las informaciones. Todo desarrollador web está confrontado con la creación de interfaz.

Voy a tratar de seguir algunas reglas ergonómicas. De un punto de vista general, el sitio y las paginas deben ser bien organizados.

- los elementos de navegación son fáciles a buscar,
- los menús ayudan al internauta que navega en los contenidos sin perder tiempo
- se debe evitar el exceso de información (ruido visual),
- las paginas deben ser sencillas, sin sobrecarga inútil
- facilite el acceso a los elementos clicables (talla, posición),
- Las páginas deben seguir la misma organización y diseño general para que el usuario no se pierda,
- el usuario debe saber donde se encuentra a cada instante.

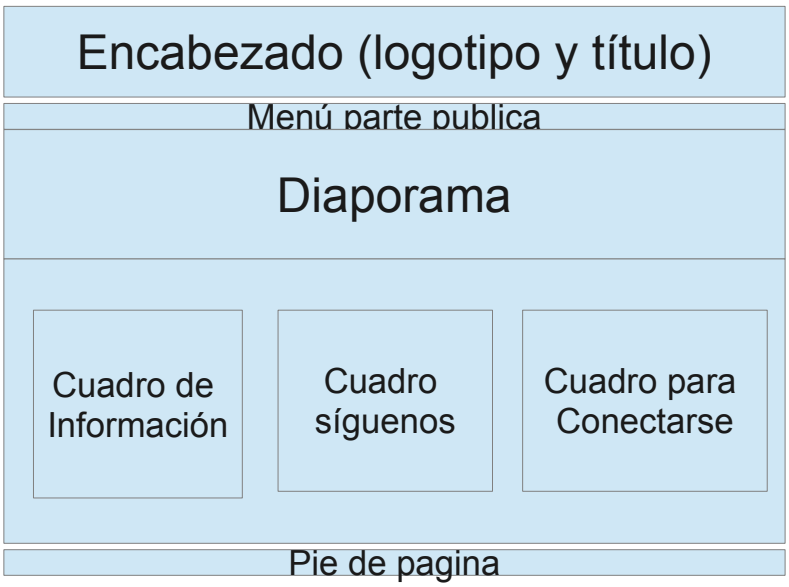
En el caso de interoperatividad entre el usuario y el sitio web, la gestión de los errores de seguir estas reglas:

- hacer todo para que el internauta sepa lo que debe hacer,
- minimizar los errores gracias a las etiquetas y gracias a las leyendas de los campos,
- evitar los errores gracias a la talla de los campos y al tipo de elementos de formulario,
- pedir una confirmación para las acciones a riesgos,
- facilitar la localización de los errores,
- dar una explicación precisa del error (evite los mensajes del tipo " Un error sobrevino "),
- El internauta debe fácilmente poder corregir sus errores (poder corregir varios errores a la vez).

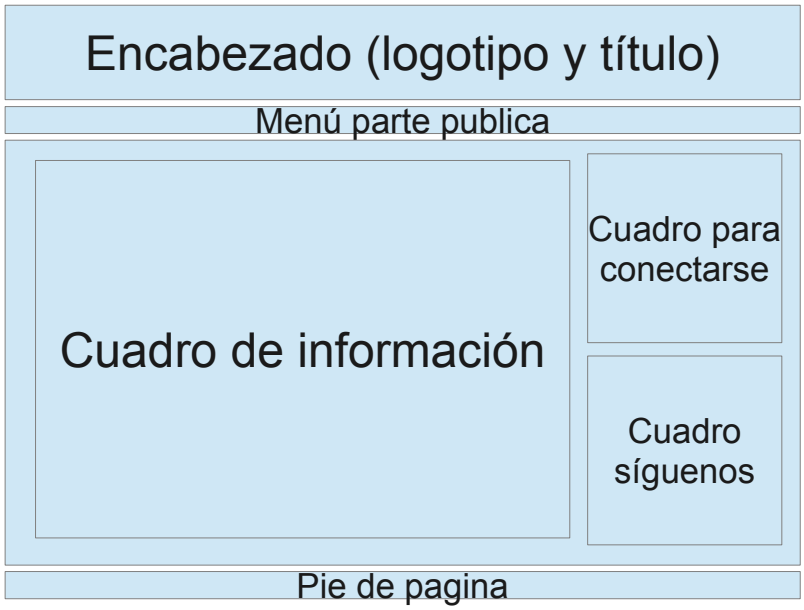
Interfaz parte pública

Para la parte pública tenemos 2 formatos de página : la página principal (1) y las paginas de informaciones generales (2).

Parte pública (1) Organización de la página principal.



Parte pública (2) Organización de las paginas fijando informaciones.



Interfaz parte privada

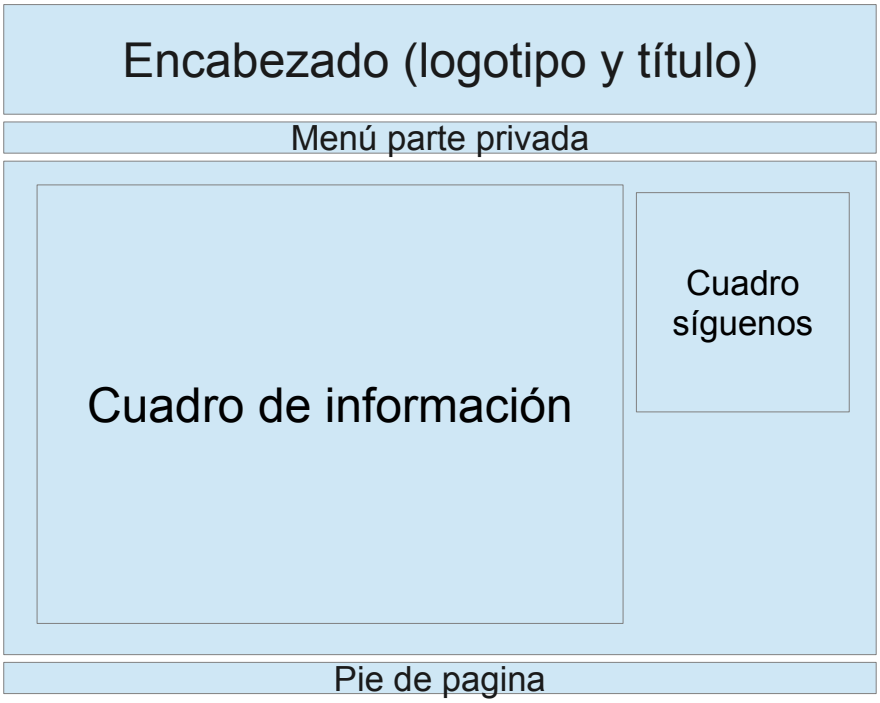
Para la parte privada tenemos 4 formatos de página distintos:

- paginas para acceder a las informaciones (3) (noticias, descargas y proyectos)

3 formatos de paginas para la parte de administración:

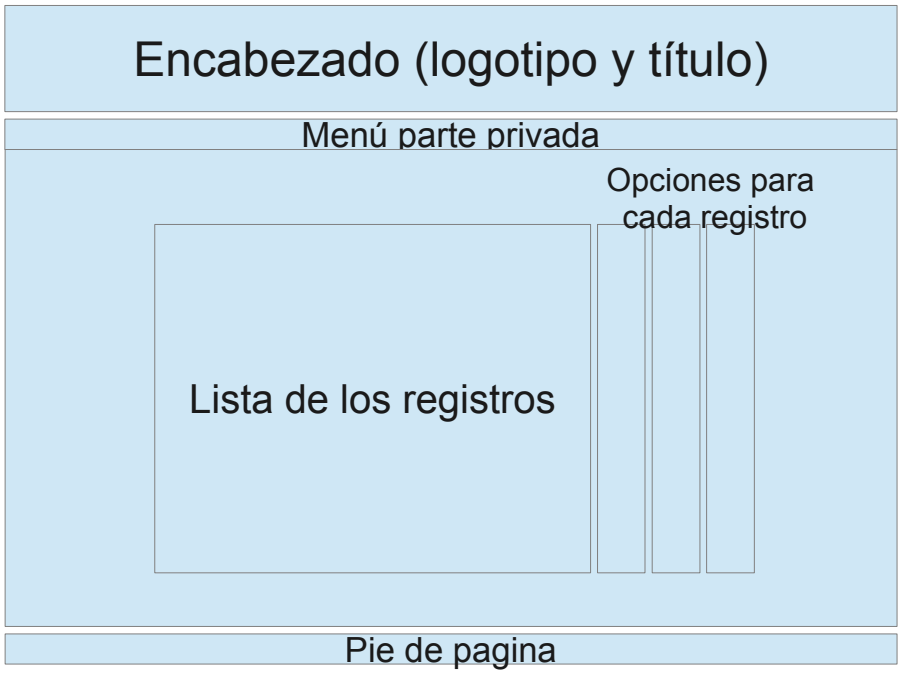
- lista de los registros que el administrador puede manejar de acuerdo a su perfil (4),
- formulario para crear un registro o modificar los datos (5),
- página para fijar los mensajes resultado de una operación (6) (crear, modificar o borrar)

Parte privada (3) Organización de las paginas fijando informaciones. Permitirá de fijar las noticias, la lista de las descargas y los datos de los proyectos por país.



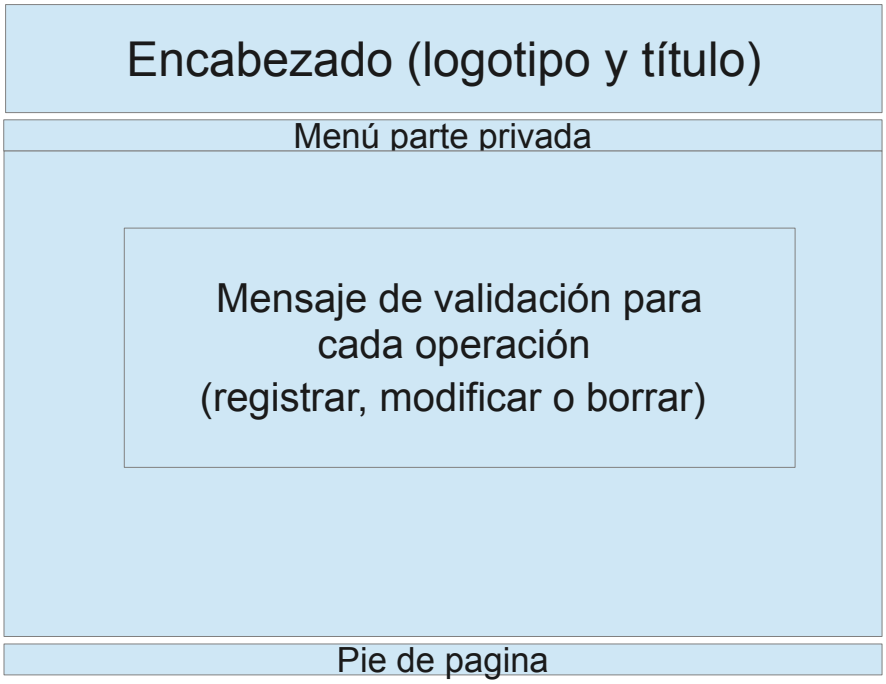
Parte privada (4) Organización de las paginas permitiendo el manejo de los registros por los usuarios que tendrán derechos de administración.

Se trata de los registros para : las cuentas de usuarios, las tablas de referencia, las noticias, las descargas, los datos de los proyectos por país y de los flujo RSS.



Parte privada (5) Organización de las paginas donde tendremos los campos de los formularios para registrar y/o modificar datos.

Parte privada (6) Organización de la pagina fijando los mensajes de validación de las operaciones.



2.2.3 Plan de prueba

El plan de prueba es una guiá a seguir para definir si el sistema cumple con los requisitos requeridos. Para cada perfil, se debe verificar si accede a los recursos y informaciones establecidos.

El plan de prueba permite :

- definir las etapas del proceso de prueba,
- encontrar el máximo de anomalías,
- identificar los problemas importantes y evaluar los riesgos asociados,
- validar un nivel de disponibilidad y la robustez esperado por el producto,
- validar la adecuación del producto con los requisitos.

El sitio web se compone de 2 partes, una pública y una privada, y de varios sub-sistemas. El plan de prueba toma en consideración este recorte permitiendo de utilizarlo este plan de prueba para diferentes niveles:

- Pruebas unitaria : verificación que cada componente funciona.
- Pruebas de integración : validación de la interacción entre sub-sistemas.
- Pruebas de implantación : validación del sistema en el entorno de producción.

- Pruebas de aceptación : los usuarios validen el funcionamiento global del sistema.

En los anexos, se encuentra el detalle de las pruebas para cada parte del sitio. Se presenta debajo la forma de tablas con un acción a realizar y dos columnas par la validación y los comentarios. Ejemplo para la parte publica.

Parte pública

Cualquier usuario debe :

	Validación	Comentarios
Acceder al sitio		
Seleccionar su idioma preferida		
Acceder a todas las paginas del menú Sidunea		
Acceder a todas las paginas del menú Proyectos		
Acceder a todas las paginas del menú Enlaces		
Desde el menú Inicio regresamos a la pagina principal		
Enlaces sobre SIDUNEA World World funciona		

2.3 Diseño

En este capitulo, vamos a basarnos sobre el análisis anterior para definir los modelos y especificaciones del diseño del sistema, cubriendo los parágrafos siguientes :

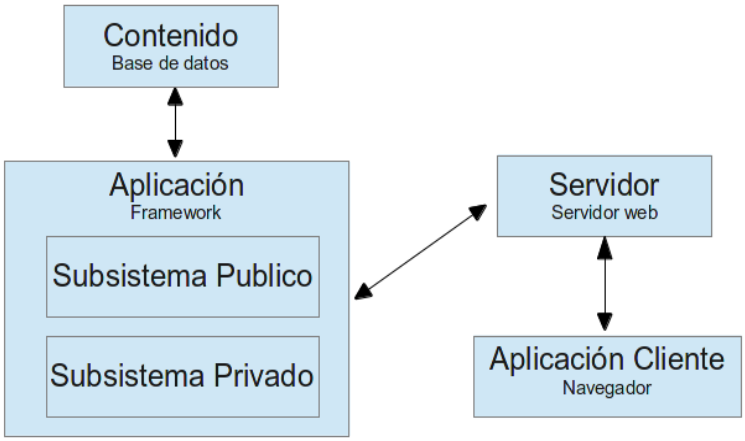
- **Arquitectura del sistema :** la definición de la arquitectura nos dará una visión de los diferentes componentes del sistema. Gracias a esta visión global tendremos materia para profundizar el diseño. La descomposición en subsistemas simplificará el desarrollo y facilitará el mantenimiento.
- **Elección de software :** se definirá la lista de los programas, herramientas, lenguajes de programación que utilizaremos llevar a cabo el desarrollo del sistema.
- **Requisitos de implantación** : Se establecerá los requisitos para la implantación del sistema en el ambiente de producción. Se debe definir el entorno tecnológico y los materiales necesarios para la implantación del sistema cuando trabajará en entorno real.

2.3.1 Arquitectura del sistema

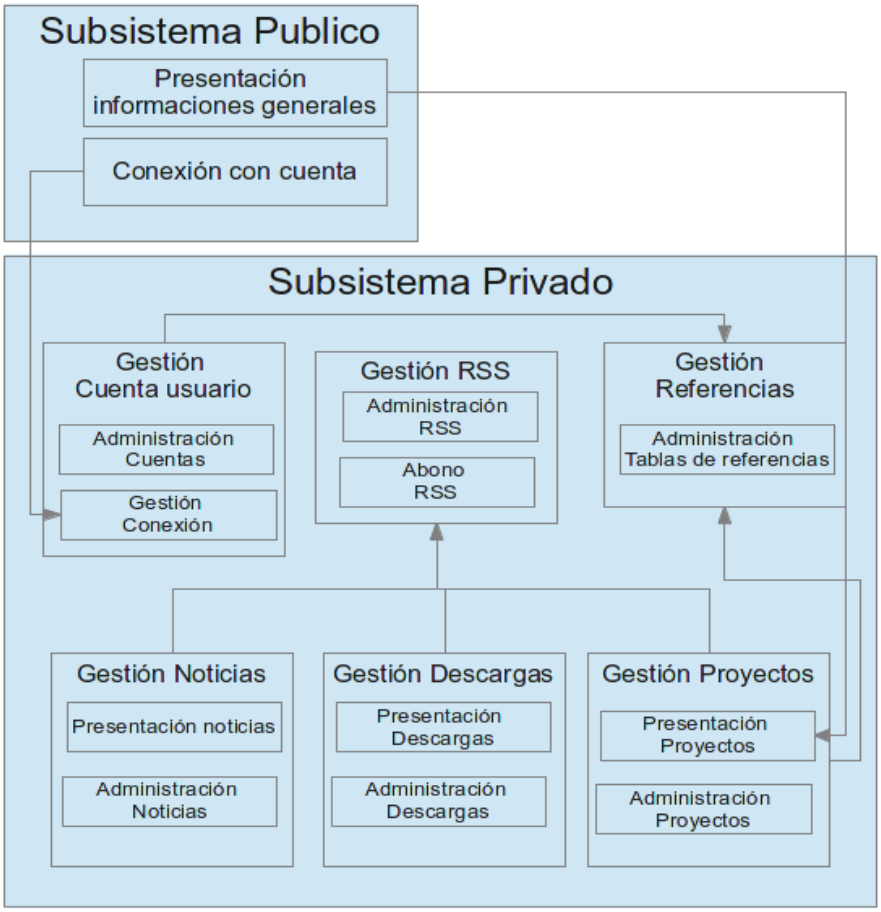
La definición de la arquitectura se basara sobre la identificación de los componentes del sistema. La descomposición en subsistemas simplificará el desarrollo y facilitará el mantenimiento. Vamos a ver a arquitectura conceptual y la arquitectura física en este capitulo.

Componentes del sistema

Tenemos 4 componentes principales : la base de datos, la aplicación, el servidor web y la parte cliente



Detallemos la parte de la aplicación donde se ubica el subsitema publico y subsistema privado.



El subsistema publico llama sub-grupos del sistema privado:

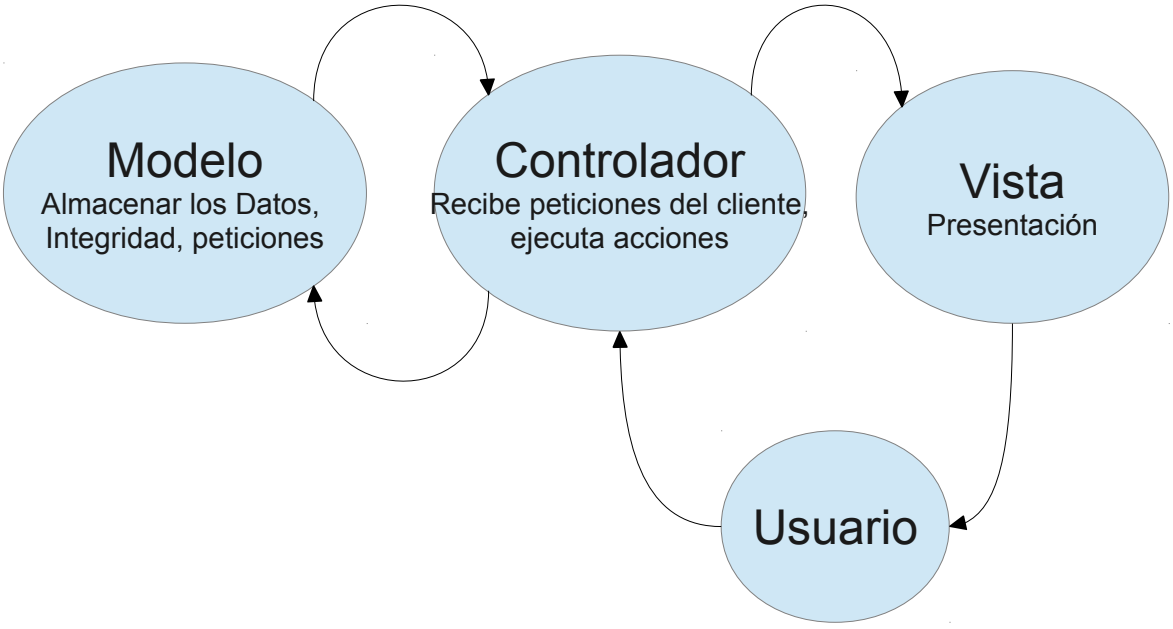
- una llamada para la conexión de los usuarios,
- una llamada al sub-grupo Presentación Proyecto en la medida que fijamos una parte de los datos de los proyectos en la parte pública del sitio.

En el subsitema privado las principales interacciones son entre :

- los sub-grupos Gestión Cuenta Usuario y Gestión Proyectos que utilizan las tablas de referencias,
- la Gestión de noticias, descargas y proyectos que interaccionan con la Gestión RSS

Arquitectura conceptual - Modelo MVC - Modelo-Vista-Controlador

El patrón Modelo-Vista-Controlador es una arquitectura de diseño software para separar los componentes de aplicación en tres niveles : interfaz de usuario, lógica de control y lógica de negocio.



El **modelo** representa el corazón de la aplicación: procesamientos de datos, interacciones con la base de datos, etc. Describe los datos manipulados por la aplicación. Reagrupa la gestión de estos datos y es responsable de su integridad. La base de datos será uno de sus componentes. El modelo contiene métodos estándares para actualizar estos datos (inserción, supresión, cambio de valor). Ofrece también métodos para recuperar estos datos. Los resultados devueltos por el modelo no se ocupan de la presentación.

La **vista** es con que el usuario interactúa precisamente. Su primera tarea es presentar los resultados devueltos por el modelo. Su segunda tarea es recibir toda acción del usuario (clic de ratón, selección de un botón radio, entrada por texto, etc.). Estos diferentes acontecimientos son enviados al controlador. La vista no efectúa tratamiento, se contenta con fijar los resultados de los tratamientos efectuados por el modelo y de interactuar con el usuario. La vista puede dialogar directamente con el modelo en caso de consulta de datos para fijarlos.

El **Controlador** se encarga de la gestión de los acontecimientos de sincronización para actualizar la vista o el modelo y sincronizarlos. Recibe todos los acontecimientos del usuario y engancha las acciones que hay que efectuar. Si una acción necesita un cambio de los datos, el controlador pide la modificación de los datos al modelo, y este último notifica a la vista que los datos cambiaron para que se actualice. Ciertos acontecimientos del usuario no conciernen a los datos sino la vista.

En este caso, el controlador pide a la vista de modificarse. El controlador no efectúa ningún tratamiento, no modifica ningún dato. Analiza la demanda del cliente y se contenta de llamar el modelo adecuado y de devolver la vista que corresponde a la demanda.

Arquitectura física

La arquitectura más común para un sitio web es la arquitectura 3 tercios. Se compone de :

- **Capa de presentación** : es la capa de presentación de un sitio o aplicación web. Corresponde a la parte de la aplicación visible e interactiva con los usuarios. (puesto cliente con navegador)
- **Capa de negocio** : recibe las peticiones de los usuarios, manejar las reglas de negocio del aplicativo, se encarga de pedir los datos a la capa de datos y restituye los resultados a la capa de presentación. Se compone de un servidor web y/o de un servidor de aplicación.
- **Capa de datos** : Consiste en la parte manejando el acceso a los datos del sistema. Físicamente es un servidor de base de datos que representa el último elemento clave de una infraestructura Web. Almacenan las informaciones de los sitios o aplicaciones web restituyendo los datos con arreglo a las necesidades aplicativas. Los servidores de base de datos están en el corazón de los servicios Web evolucionados.

Nuestro tarea es la de ayudar los países a implementar Sidunea World para manejar su administración aduanera. No es la de administrar servidores y arquitectura de red porque no tenemos los recursos para mantener un espacio con un equipo encargo de varios servidores aunque necesitamos servidores para nuestras aplicaciones.

En el caso de este sitio web, vamos a utilizar un servicio de alojamiento para nuestra aplicación.

El tipo de alojamiento sera debajo un alojamiento compartido, las razones son :

- costo bajo,
- no tenemos elementos multimedia,
- no se necesita muchos recursos,

- el numero de visitas por día sera bajo,
- la evolución del numero de visita no cambiara,
- conocemos el numero de usuarios (entre 250 – 500).

Existe numeroso servicios de alojamiento que suministran casi los mismo servicios. Desde hace año tenemos un nombre de dominio en Inmotion Hosting. Los servicios dados por este empresa son suficientes para el sitio que queremos implementar. Se puede consultar la oferta con este enlace :

www.inmotionhosting.com/business-hosting

El plan que elegimos, Small Business Starter, tiene como características principales : espacio de disco ilimitado, cantidad ilimitada de transferencia de datos por mes, copia de seguridad gratis, 2 bases de datos, soporte técnicos 24x7x365, una selección alrededor de 300 aplicaciones a instalar.

Los servidores son instalados con sistema operativo Linux versión 2.6.18-348.16.1.el5 con una arquitectura x86_64.

2.3.2 Elección de software

Lista de los programas y sus licencias

Componente	Programas / lenguaje	Versión	Licencia
Interprete de script	PHP	5.3.27	PHP 5 are distributed under the PHP License v3.01
Servidor web	Apache	2.2.26	Apache License - http://www.apache.org/licenses/LICENSE-2.0
Base de datos	MySQL	5.5.32-cll	Since June 2000 (that is, since version 3.23.19) the *GNU Public License* (GPL) has been valid for MySQL
Librería javascript	JQuery	1.4.2	MIT License https://github.com/jquery/jquery/blob/master/MIT-LICENSE.txt
Sistema de administración de control de versión	Subversion edge	4.0.0	Subversion Edge is dual licensed under the open-source GNU Affero General Public License v3 (AGPLv3) and a commercial

			CollabNet license for partners that would like to integrate Subversion Edge into their proprietary offerings.
Sistema de control de versión	Svn	1.8.0	Apache License Version 2.0 - http://svn.apache.org/repos/asf/subversion/trunk/LICENSE
Entorno de desarrollo	Aptana Studio 3	3.4.2	Aptana Studio is licensed under the terms of the GNU Public License (GPL) v3 (with exceptions) http://www.aptana.com/legal/
Framework PHP	Codeigniter	2.1.4	Licencia Codeigniter http://ellislab.com/codeigniter/user-guide/license.html

2.3.3 Requisitos de implantación

En el apartado de *2.3.2 Elección de software* tenemos la lista de los programas y herramientas que utilizaremos para el desarrollo del sistema. Las versiones son dictadas por el servicio de alojamiento que hemos seleccionado. Debemos seguir lo más cerca posibles la instalación de las mismas versiones en el entorno de desarrollo para estar seguro que lo que haremos sea compatible con el entorno de producción. Eso es necesario para :

- la base de datos (MySQL),
- el interprete de script (PHP),
- el servidor web (Apache),
- el framework (Codeigniter)

El entorno de desarrollo sera instalado sobre un ordenador portátil con las características siguientes :

- Memoria : 8 Gb
- Disco : SATA 750 Gb
- CPU : Interl Core-i5-2430M, 4 cuerpos
- Distribución Linux : Debian 6.0
- Kernel : 3.2.0-4-686-pae

3 EJECUCIÓN

3.1 Desarrollo

En este capítulo se presenta el desarrollo del proyecto que implica la presentación de la planificación de las actividades a realizar para cumplir con los requisitos del sistema a desarrollar y los aspectos técnicos del proyecto.

- **Planificación de las actividades :** en este parágrafo se identificará las tareas a realizar y se estimará la duración de las actividades para presentar el plan de trabajo. Un cronograma de las actividades (diagrama de Gantt) será nuestro soporte para presentar la planificaciones de las actividades.
- **Desarrollo del proyecto :** nos apoyaremos sobre el estudio de viabilidad, el análisis y la descripción del diseño para llevar a cabo el desarrollo del proyecto.

3.1.1 Planificación de las actividades

Prever las tareas y planificarlas es indispensable para llevar a cabo un proyecto de desarrollo cualquiera que sea. Las grandes etapas del desarrollo de este proyecto son:

- definición y preparación del entorno de desarrollo,
- desarrollo del proyecto,
- implantación en entorno de producción.

El diagrama de Gantt nos presenta el detalle de las tareas que debemos llevar a cabo para lograr el desarrollo del proyecto. Al fin de cada módulo o parte desarrollado, tenemos un hito que nos dará información sobre el avance del proyecto y podremos ajustar las fechas de entrega en caso de desviaciones. (Anexo 3)

Debido a un cambio de planificación muy fuerte en mi trabajo no pude dedicar tiempo para el proyecto entre el 6 de noviembre de 2013 y el 2 de diciembre de 2013. El resultado de este imponderable fue un cambio de planificación que tenia y en consecuencia hubo que reducir el tiempo a dedicar para cada tarea. Eso impactó las siguientes tareas de desarrollo :

- tablas de referencias,
- gestión de noticias,
- gestión de los proyectos,
- gestión de los SPR. (System Problem Report)

Después de una consulta con mi tutor, decidimos de no desarrollar la gestión de los SPR (System Problem Report) debido a que ahora casi no se utiliza. Además el foro técnico de SIDUNEA World, que abrió hace poco tiempo, reemplazó este pesado procedimiento,

Al nivel de nuestra análisis de riesgo, nos ubicamos en la categoría : Riegos Humanos - Re-afectación de recursos sobre otras prioridades operacionales.

Una nueva planificación de las tareas se hizo reduciendo los tiempos de desarrollo y en consecuencia necesité aumentar las horas de trabajo por día. Una reducción de

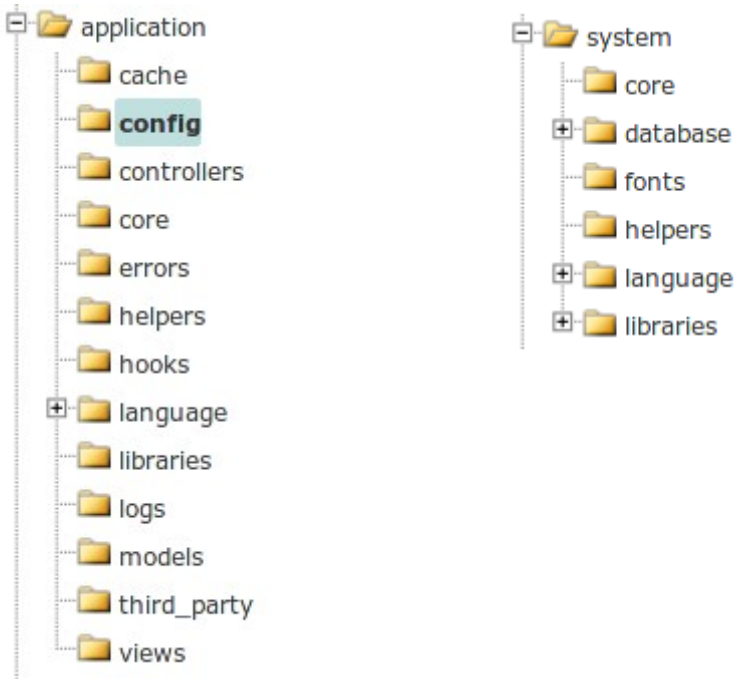
casi 30% del tiempo del proyecto puede afectar la calidad del trabajo, aumentar el estrés del equipo entre otro. Se puede consultar el plan de trabajo revisado en el Anexo 4.

3.1.2 Desarrollo del proyecto

En este capítulo, veremos puntos relevantes del desarrollo del proyecto.

Codeginiter

Una vez instalado Codeigniter se presenta debajo una arquitectura en dos partes : system y application.



Dentro las carpetas system encontraremos el corazón del framework Codeigniter. Todas las clases, drivers, librerías se ubican aquí.

La parte aplicación está reservada para el desarrollo de una aplicación o de un sitio web. Podemos entender sin dificultad que el modelo Modelo-Vista-Controlador se desplegará dentro de las carpetas models, views y controllers respectivamente.

La configuración de Codeigniter se ubica dentro la carpeta application/config. Tres archivos son importantes:

- config.php : son elementos de configuración globales que encontraremos en este archivo. La variable más importante es la definición de nuestra URL de base. Sobre la máquina de desarrollo con una instalación por defecto del servidor Apache y una instalación de Codeigniter dentro la carpeta /var/www/asyamer, la definición de nuestra URL de base es la siguiente :

```
$config['base_url']	= 'http://localhost/asyamer';
```

- database.php : es el lugar para definir la conexión a la base de datos. Nombre del host, nombre del usuario, contraseña, nombre de la base de datos, puerto de conexión. Son las variables mínimas para configurar de la conexiona a su base de datos.

```
$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'uoc';
$db['default']['password'] = 'uoc123';
$db['default']['database'] = 'asyamer';
$db['default']['dbdriver'] = 'mysql';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = FALSE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;
$db['default']['port'] = 3306;
```

- Autoload.php **:** nos permite cargar clases, helper, modelos, librerías, archivos de traducciones entre otros de manera automática. Por este medio estos recursos estarán disponibles globalmente para todo el sitio . Necesitaremos las clases database y session. Su carga automática se define de la manera siguiente :

```
$autoload['libraries'] = array('database','session');
```

Con esta configuración mínima estamos listo para empezar el desarrollo del proyecto.

El controlador

Debemos paramos un poco sobre el controlador porque es el corazón del modelo modelo-vista-controlador. El controlador recibe las peticiones del usuario, llama a las funciones del modelo para obtener datos de la base de datos y construye la vista para fijar el resultado. Con un ejemplo vamos a entender la filosofía de Codeigniter.

```
<?php
class Pages extends CI_Controller {
        public function display()
        {
                $this->load->view('myview');
        }
}
```

Este controlador fija la página myview.php. El archivo myview.php se ubica en la carpeta application/views y puede contener código html como código php.

Para llamar este pagina desde el navegador debemos tener la URL siguiente:

```
http://localhost/Pages/display
```

El formato del URL se compone primero del nombre del controlador y después una función que se encarga de fijar nuestra página. El formato genérico es el siguiente :

```
http://example.com/[controller-class]/[controller-method]/[arguments]
```

El último elemento de esta cadena está reservado para los argumentos que queremos pasar a nuestra función. Si nuestra función ahora requiere parámetros podemos pasarlos por el URI. El controlador tendrá en código siguiente :

```
<?php
class Pages extends CI_Controller {
        public function display($id,$myplan)
        {
                $data['id'] =  $id;
                $data['myplan'] =   $myplan;
                $this->load->view('myview',$data);
        }
}
```

y llamaremos nuestra vista de esta manera :

```
http://localhost/Pages/display/12/15
```

La función display del controlador Pages recibirá directamente los valores de los argumentos pasados en el URI. Aproveché de implementar en este ejemplo como pasar datos a la vista gracias a la tabla $data.

El controlador es el componente central de Codeigniter.

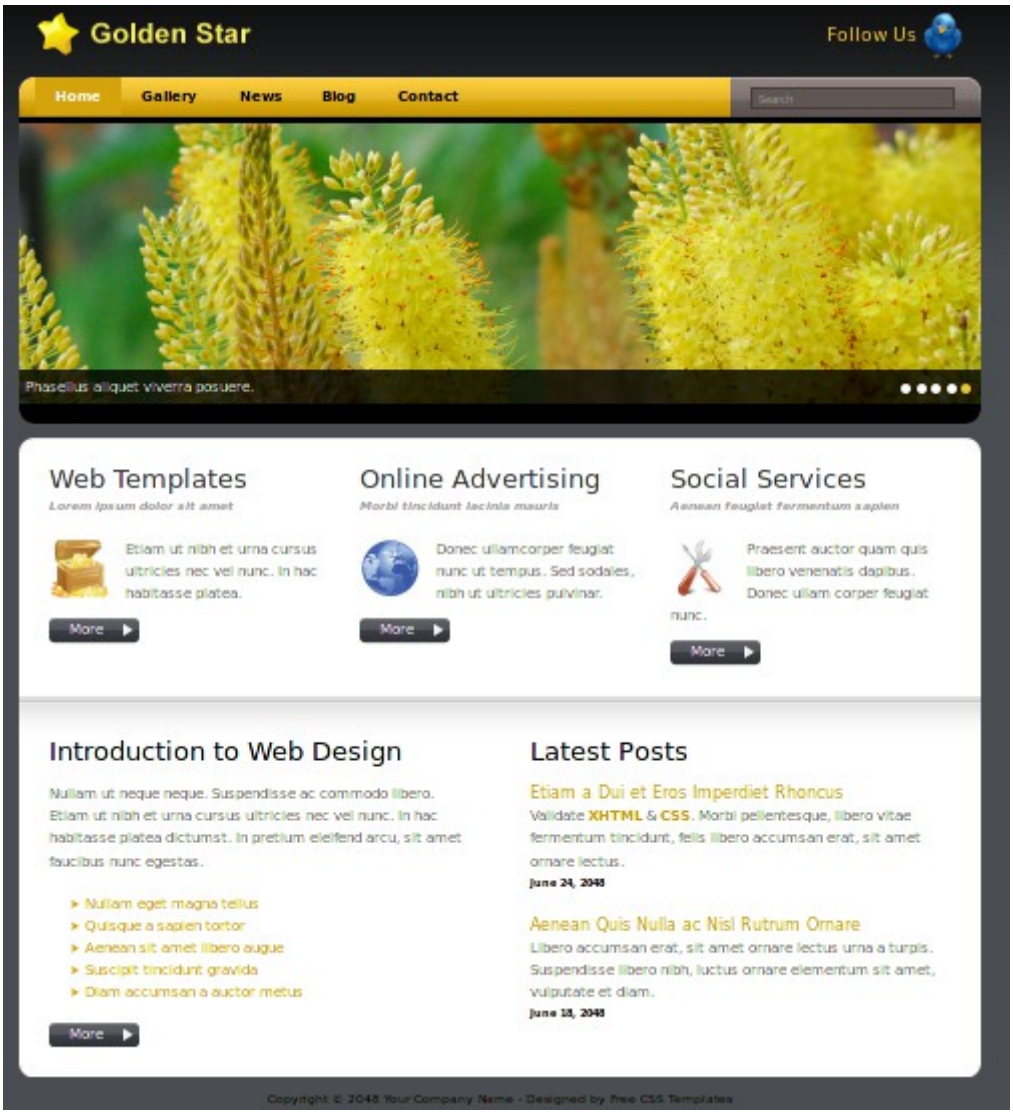
CSS

De acuerdo a nuestro análisis, hemos definido el diseño requerido para nuestras páginas que sea para la parte pública o privada. Dos posibilidades se presentan:

- diseñar desde cero una hoja de estilo.
- buscar en Internet una hoja que se acerque al diseño requerido y adaptarla.

Elegí la segunda opción. Las razones son que no tenía experiencia en CSS y apoyarme sobre un ejemplo facilitaría el aprendizaje de las hojas de estilo.

Encontré un en el sitio http://www.templatemo.com/ que se acercaba a lo que quería con una licencia libre de modificación, utilizable para sitos web personales o comerciales y sin restricción.

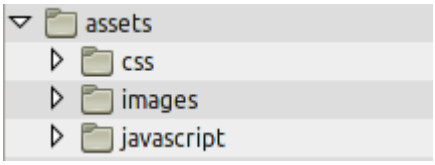


La primera tarea fue de integrar esta hoja de estilo en la arquitectura Codeigniter de manera que se podía mostrar en a pantalla. Esta hoja de estilo se compone de 3 carpetas:

- ***css*** : donde se encuentra los diferentes hojas de estilos para la presentación de la paginas y del diaporama,
- ***images*** : donde se encuentra todas las imágenes utilizadas,
- ***js*** : donde se encuentra las librerías javascript para utilizadas para el diaporama.

Esta hoja de estilo tiene también páginas html de ejemplos que corresponde a cada uno de los elementos del menú: index.html, blog.html, gallery.html, news.html y contact.html

En la arquitectura de Codeginiter, creé una carpeta denominada ***assets*** para almacenar estas carpetas con la arquitectura siguiente :



Para probar la página principal index.html modifiqué la parte head de manera que podía alcanzar la carpeta assets y sus archivos. Ahora hacer referencia a uno de los archivos de javascript se hace de la manera siguiente:

```
<script src="http://localhost/asyamer/assets/javascript/jquery.min.js"
type="text/javascript"></script>
```

Helper

Además de las clases que podemos desarrollar para extender las funcionalidades de Codeigniter, los helpers nos permiten almacenar una colección de funciones dedicadas a una tarea particular. Codigniter suministra varios helpers como por ejemplo Form Helper que nos ayuda a crear formularios.

Aproveché de esta facilidad ofrecida por CodeIgniter, desarrollando un helper para simplificar la referencia a los archivos de la carpeta assets que sea para las hojas de estilos, los archivos de javacript y todas las imágenes. Tres funciones componen este helper:

- js_url($nom) : para hacer referencia a un archivo javascript de la carpeta assets/javascript. El parámetro $nom corresponde al nombre del archivo javascript que queremos utilizar.
- css_url($nom) : para hacer referencia a una hoja de estilo de la carpeta assets/css. El parámetro $nom corresponde al nombre de la hoja de estilo que queremos utilizar.
- img($nom, $alt = '') : para hacer referencia a una imagen de la carpeta assets/images. El parámetro $nom corresponde al nombre de la imagen que queremos utilizar y el paramtro $alt especifica un texto alterno para una imagen, si la imagen no puede ser mostrada.

Tomando el ejemplo anterior de la referencia al archivo jquery.min.js ahora gracias a este nuevo helper tenemos el código siguiente :

```
<script src="<?php echo js_url('jquery.min'); ?>" type="text/javascript"></script>
```

Los helpers suministrados por Codeigniter se ubican en las carpeta system/helpers con el formato de archivo siguiente : nombre_helper,php.

Form es el helper para implementar los formularios, se encuentra en el archivo form_helper.php

Los helpers desarrollados para extender una aplicación se ubican en la carpeta application/helpers. Este nuevo helper se denomina assets_helper.php.

Si para nuestra aplicación queremos utilizar un helper debemos cargarlo al momento que lo necesitamos, según las convenciones dentro un controlador , ejemplo para nuestro helper assets :

```
$this->load->helper('assets');
```

Tenemos la posibilidad de cargar un helper para toda la aplicación de manera automática por el archivo application/config/autoload.php, ejemplo para nuestra aplicación :

```
$autoload['helper'] = array('url','assets');
```

Clase MY_Controller

Según la definición de un controlador por Codeigniter, un controlador es una clase que será asociada a un URI (Uniform Resource Identifier). Los controladores son el corazón de la arquitectura modelo-vista-controlador. Cuando un usuario llama a una página, se hace referencia a el controlador que se encargará de llamar la vista de la página requerida. En el proyecto, la página del inicio se llama por el controlador c_welcome.

Un controlador en Codeigniter es la extensión de la clase principal de los controladores que se llama CI_Controller. En lenguaje orientado a objeto esta herencia hace que el controlador hereda de todos las funciones y atributos de la clase pariente.

En el proyecto, como las páginas de la parte pública tienen la misma arquitectura, he desarrollado una clase MY_Controller que será el controlador principal de las páginas del sitio. Esta clase se encarga de lo siguiente :

- cargar los archivos de traducciones de acuerdo a la selección del idioma por el usuario almacenado en sus datos de sesión. Se hizo sobrescribiendo el constructor de la clase principal de los controladores de Codeigniter,
- a la primera carga de la página principal por defecto el idioma del sitio es el inglés. (función initializeData())
- la función lang() se llamará de cada página para guardar el idioma seleccionado en la sesión del usuario. Llamará al controlador de la pagina requerida para mostrarla con el idioma seleccionado.
- La función index_view($controllerName,$viewPage) contiene el código permitiendo mostrar las páginas de la parte pública llamando varias vistas :

- head : parte que contiene el código de los tags <head>.
- pageheader : para mostrar el encabezado de la página y las opciones para las idiomas.
- menu : para mostrar el menú.
- $viewPage : el contenido de la pagina
- right_col : la parte derecha de una página que se encarga del cuadro de conexión y del cuadro síguenos.
- footer: pie de página.

  La función index_view se llamará de cada controlador con los parámetros siguientes:

  - $viewName : nombre del controlador requerido.
  - $viewPage : nombre de la vista requerida.

La clase MY_Controller se ubica en la carpeta application/core.

Arquitectura de un controlador de una página de la parte pública

Detallemos un controlador de una página estática.

```
class c_Asycuda_Tariff extends MY_Controller {
        public function index(){
                $this->load->helper(array('form'));
                $this->indexView('c_asycuda_tariff','asycuda/v_tariff');
        }
        public function lang_fr(){
                $this->lang('french','c_asycuda_tariff');
        }
        public function lang_en(){
                $this->lang('english','c_asycuda_tariff');
        }
        public function lang_es(){
                $this->lang('spanish','c_asycuda_tariff');
        }
}
```

Comentarios del código:

- el controlador es una extensión de nuestra clase para los controladores MY_Controller

- la función index(), se llama por defecto cuando se ejecuta el código del controlador, es una característica de Codeigniter, y se encarga :,
  - de cargar el helper Form porque lo necesitamos para manejar el formulario para la conexión del usuario a la parte privada,
  - de llamar la función index_view de la clase MY_Controller para mostrar la página requerida,
- las funciones lang_fr(), lang_es() y lang_en() llaman la función lang() de la clase MY_Controller después de la selección de un idioma.

Traducción

Codeigniter nos ofrece una arquitectura para traducir contenidos. La clase language se encarga de esta tarea. Primero, debemos crear una carpeta para cada idioma en el directorio application/language. Dentro cada carpeta de idioma tendremos archivos conteniendo las traducciones. En estos archivos definiremos clave para cada traducción requerida. Una clave y su traducción sigue el formato, ejemplo para la traducción de la palabra Inicio :

Dentro el archivo applicacion/language/english/menu_lang.php

```
$lang['menu_home'] = "Home";
```

Dentro el archivo applicacion/language/french/menu_lang.php

```
$lang['menu_home'] = "Accueil";
```

Dentro el archivo applicacion/language/spanish/menu_lang.php

```
$lang['menu_home'] = "Inicio";
```

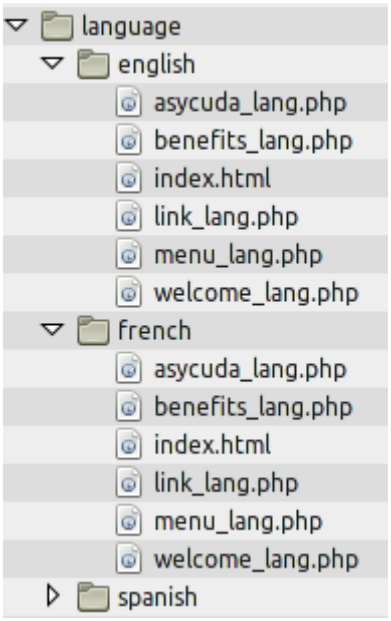
Se observa que la clave definida es la misma dentro cada archivo. Los archivos también tienen el mismo nombre y siguen el formato : nombre_lang.php

Utilizar las traducciones es muy sencillo, se necesita solamente llamar en nuestra vista la clave para obtener su traducción. Ejemplo para la clave menu_home :

```
<li class='active'><a href="<?php echo site_url('welcome/index') ?>">
        <span><?php  echo $this->lang->line('menu_home')?
></span></a></li>
```

Por supuesto se necesita una implementación que carga los archivos de idiomas adecuados al momento requerido. Ya tenemos este sistema en la clase MY_Controller.

Vista de la arquitectura de las carpetas de traducciones.



Sesión

La sesión es un mecanismo que le permite a PHP guardar en memoria un número ilimitado de valores entre varias demandas del mismo usuario. No se necesita implementar en PHP nativo el manejo de las sesiones. Codeigniter nos ofrece gracias a la clase Session una gestión automática de las sesiones dentro la base de datos. Primero para configurar las sesiones debemos cargar la clase Session cuando empieza un usuario a navegar en el sitio.

En el directorio applicacion/config/autoload.php debemos cargar la clase Session y Database, como siguie :

```
$autoload['libraries'] = array('database','session');
```

Por seguridad la utilización de las sesiones necesita una clave de cifrado. Se define dentro el archivo applicacion/config/config.php nuestra clave de cifrado :

```
$config['encryption_key'] = 'hi02vqkKc641ikFPU59V7AyEi9R0cNzv';
```

Las últimas configuraciones son para definir si nuestra aplicación utiliza una base de datos para las sesiones y el nombre de la tabla. En el archivo applicacion/config/config.php

```
$config['sess_use_database']    = TRUE;
$config['sess_table_name']      = 'asy_sessions';
```

Por supuesto, se debe crear una tabla para recibir los datos de las sesiones. En esta tabla se guardará :

- el id de la sesión,
- la dirección IP
- los datos del User Agent,
- un timestamp de la ultima actividad.

Ejemplo de un registro de sesión en la base de datos :

session_id	ip_address	user_agent	last_activity
ee893fcfbc9bf5a7c6e80e385c92d19f	127.0.0.1	Mozilla/5.0 (X11; Linux i686; rv:17.0) Gecko/17.0 Firefo...	1389900017

Además para la parte pública, se utiliza la sesión para guardar el idioma seleccionado. Eso se ubica en el constructor de nuestra clase MY_Controller, ejemplo dentro la funcion lang() :

```
function lang($language,$ViewName){
        $this->session->set_userdata('language',$language);
        redirect($ViewName);
}
```

Ejemplo de la columna user_data de la tabla asy_session que contiene los datos del usuario :

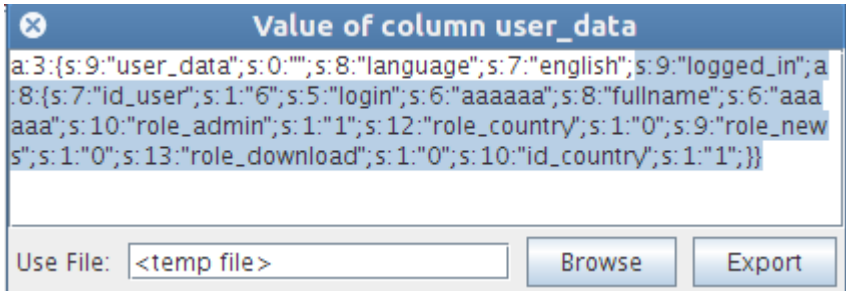
user_data
a:2:{s:9:"user_data";s:0:"";s:8:"language";s:6:"french";}

Luego en la parte privada, cuando el usuario se conecta, utilizamos la sesión para guarda otros datos. Estos datos nos sirven para conocer sus permisos y por supuesto si está todavía conectado cuando está navegando en la parte privada.

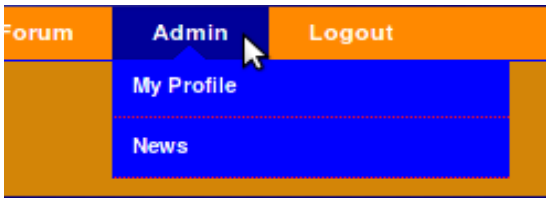
Es en el controlar verifylogin que almacena datos del usuarios en la sesión :

- id_user : el id de la cuenta
- login : el nombre de conexión
- fullname : el nombre del usuario
- role_admin : si tiene permiso de administrador
- role_country : si tiene permiso de administrador de proyecto
- role_news : si tiene permiso de administrador de noticias
- role_download : si tiene permiso de administrador de descargas
- id_country : el país del usuario

Ejemplo de la columna user_data de la tabla asy_session que contiene los datos de un usuario conectado. Se puede ver el array logged_in:



Los datos role_xxx nos sirven para adaptar el menú de acuerdo a sus permisos. Ejemplo del menú para un administrador y del menú para un administrador de noticias.



Estos datos están almacenados en la sesión dentro un array denominado : logged_in.

Cada vez que un usuario hace una petición para acceder a una página de la parte privada verificamos que en la sesión existe esta la tabla logged_in. Ejemplo en el controlador, c_news.php cuando se llama la lista de las noticias:

```
function news_display()
{
	// check if user connected
	 if($this->session->userdata('logged_in'))
	{
		$this->load->library('table');
		$this->load->library('pagination');
		$config['base_url'] = base_url() . 'index.php/c_news/news_display';
		$config['total_rows'] = $this->m_news->get_CountNewsPublished();
		$config['per_page'] = '3';
		$config['num_links'] = '3';
		$this->pagination->initialize($config);
		$data['title'] = 'News';
		$data['records'] = $this->m_news->select_AllNewsDisplay($config);
		$menu['records'] = $this->m_menu->get_CountryGroupAndCountry();
		$this->load->view('head_NoJs');
		$this->load->view('pageheaderIntranet');
		$this->load->view('intranet_menu', $menu);
		$this->load->view('news/v_news_display',$data);
		$this->load->view('footer');
	}else{
		 //If no session, redirect to login page
		 redirect('welcome', 'refresh');
	}
}
```

En caso que no exista la tabla logged_in en la sesión , eso significa que este usuario no está conectado o que su sesión expiró.

Cuando un usuario se desconecta, borramos la tabla logged_in de la sesión. Esta implementación se ubica en la funcion logout() del controlador verifylogin :

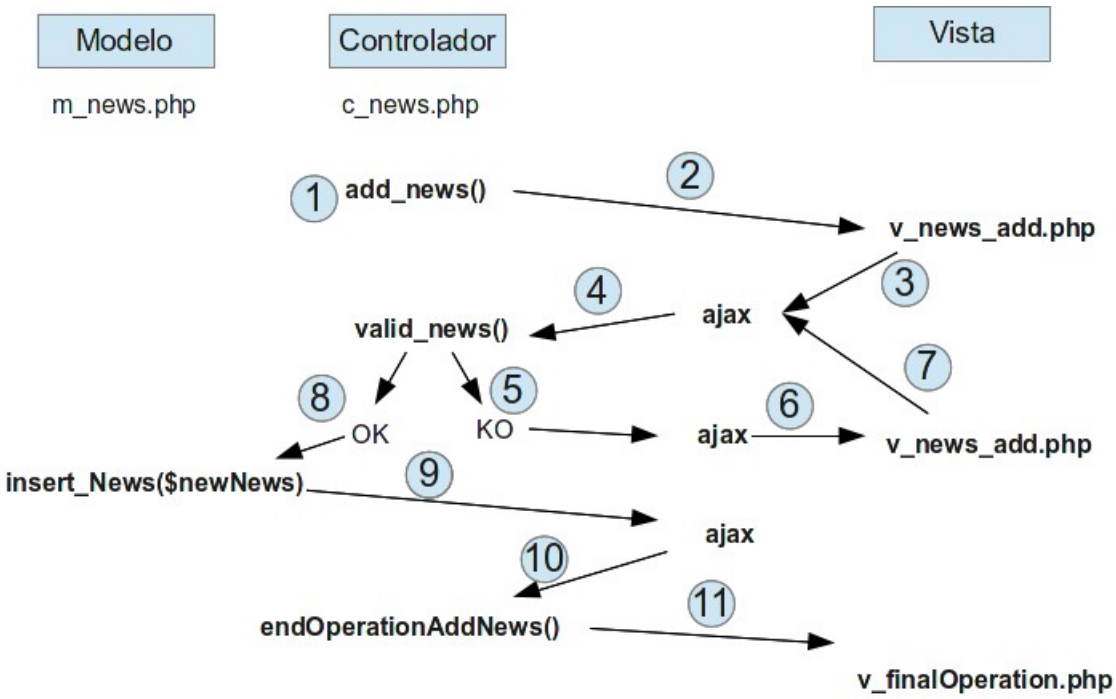
```
function logout()
{
   $this->session->unset_userdata('logged_in');
   // session_destroy();
   $this->session->sess_destroy();
   redirect('welcome', 'refresh');
}
```

Análisis del funcionamiento de un formulario

En la parte privada tenemos varios formularios para administrar los datos del sitio.

Para estos formularios, tenemos un esquema muy parecido para las operaciones de registro, modificación o supresión. Vamos a analizar el funcionamiento de un formulario para el estudio del registro de una noticia : cuales son los elementos que intervienen en cada parte del modelo MVC y los intercambios entre el navegador y el servidor.

Vista de las llamadas y intercambios del modelo MVC



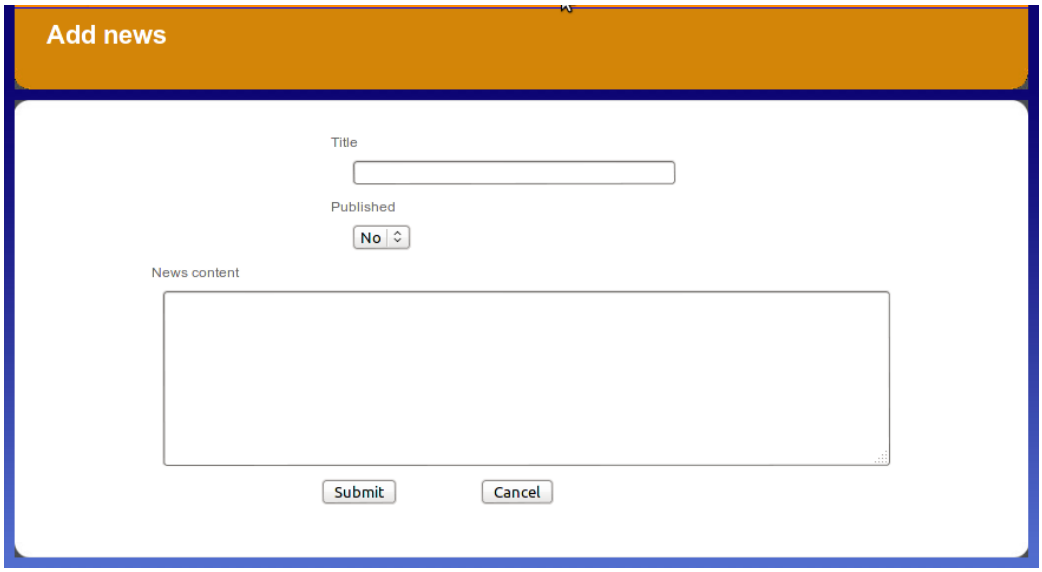
Nos situamos como un administrador de noticias.

1 – El administrador gracias al enlace Add News llama la funcion add_news() del controlador c_news.php.

2 – La función add_news() se encarga de preparar y fijar el formulario, llamando varias vistas para crear la página final. Se llama las vistas siguientes:

- news/head_news : parte que contiene el código de los tags <head>.
- pageheaderIntranet : para mostrar el encabezado de la página.
- intranet_menu : para mostrar el menú.
- news/v_news_add : en contenido de la página, aquí el formulario.
- footer : el pie de página.

El resultado de todas estas llamadas es la creación del formulario para registrar una noticia. Como una página html se compone de varias partes aprovechamos de la flexibilidad de Codeigniter para tener varias vistas que se encargan de cada una.



3 – El administrador llena el formulario y valida con el botón Submit. Esta acción llama un código javascript ubicado en news/head_news. Este código nos permite gracias a la tecnología Ajax :

- de enviar los datos del navegador hacia el servidor,
- de modificar el contenido de la página actualmente fijada de acuerdo a las respuestas del servidor, evitando así la transmisión y la fijación de una nueva

página completa.

4 – El código Ajax transmite los datos a la función valid_news() del controlador.

Esta función utiliza la clase formvalidation de Codeigniter para validar los datos del formulario. La clase formvalidation nos ofrece la posibilidad de establecer reglas de control para cada campo del formulario de acuerdo a su formato. En caso de error retornará un mensaje a la vista gracias al código Ajax.

5 – Si hay uno o varios errores, retornaremos la respuesta a la vista news/v_news_add por mediación del código Ajax.

```
$(document).ready(function() {
$('#submit').click(function() {
var form_data = {
title:$('.<?php echo 'title';?>').val(),
content:$("#content").val(),
published:document.getElementById("published").value,
ajax : '1'
};
$.ajax({
url: "<?php echo site_url($valid) ?>",
type: 'POST',
async : false,
data: form_data,
success: function(msg) {
            if (msg =='add'){
           window.location='<?php echo site_url($linkResult."/1"); ?>';}
    if (msg =='upt'){
           window.location='<?php echo site_url($linkResult."/2"); ?>';}
    else $('#message').html(msg);
}
});
return false;
});
});
```

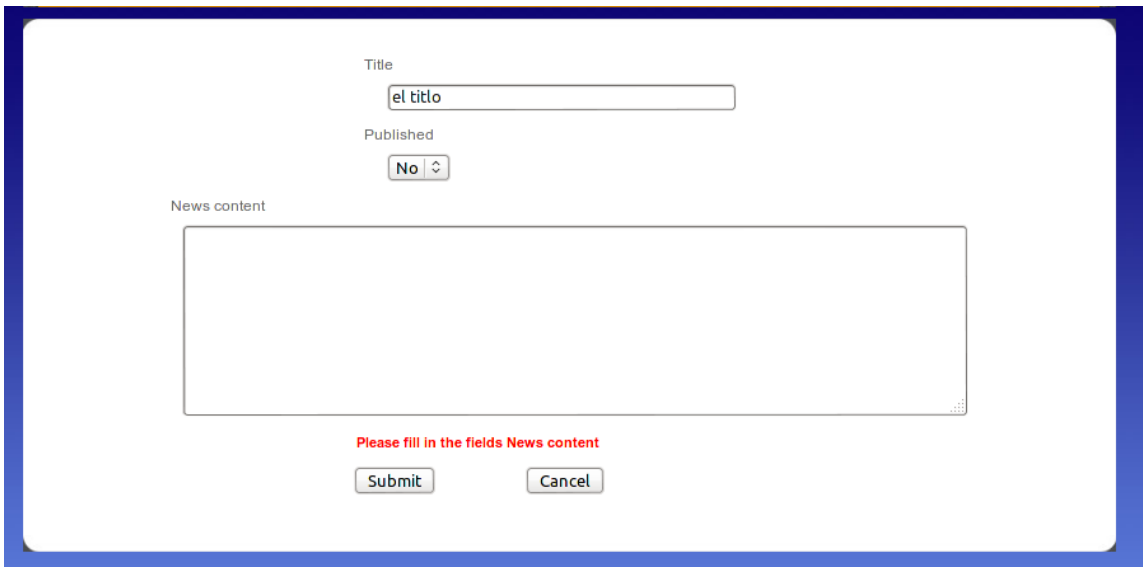
El código Ajax está a la espera de una cadena que contiene la respuesta del servidor. De acuerdo a la respuesta del servidor se aplicarán varias opciones :

- la respuesta es la cadena 'add' : no hay error, se insertó los datos en la base

de datos y se llamará la vista de fin de operación para un registro de una nueva noticia,

- la respuesta es la cadena 'upt' : no hay error, se actualizó el registro en la base de datos y se llamará la vista de fin de operación para una modificación de una noticia,
- para cualquiera otra cadena : hay un error y vamos a fijar el mensaje del error en el formulario.

6 – Ajax fija los errores en el formulario de acuerdo a las reglas de validaciones implementadas en la función valid_news() o de acuerdo a un error de la función insert_news() del modelo m_news.



7- Después la corrección y la validación por el administrador, volveremos a la etapa 4.

8 - Sin error por parte de la función valid_news(), se llama la función insert_news() del modelo m_news.php que se encarga del registro de los datos :

```
function insert_News($newNews)
{
   $res = $this->db->insert($this->table, $newNews);
          if (!$res) {
                    // database config debug = false to work
                     $msg = $this->db->_error_message();
                     $num = $this->db->_error_number();
                    if ($num == 1062){
                              $msgRes = "Title already added.Please choose another one";
                     }else{
                              $msgRes =  "Error(".$num.") ".$msg."\nContact web site administrator.";
                     }
          } else {
                    $msgRes = 'add';
          }
          return $msgRes;
}
```

En caso de suceso, retornaremos la cadena 'add', si no un mensaje de error será retornado al controlador.

La clase Active Record de Codeigniter simplifica las operaciones hacia a base de datos. Casi no se necesita implementar cadena SQL. Numerosas funciones están disponibles para cualquier tipo de petición. Ejemplos:

Inserción de un registro

```
$this->db->insert('mytable', $object);
Parámetros:
mytable : nombre de la tabla
$object : un array con los datos a insertar.
```

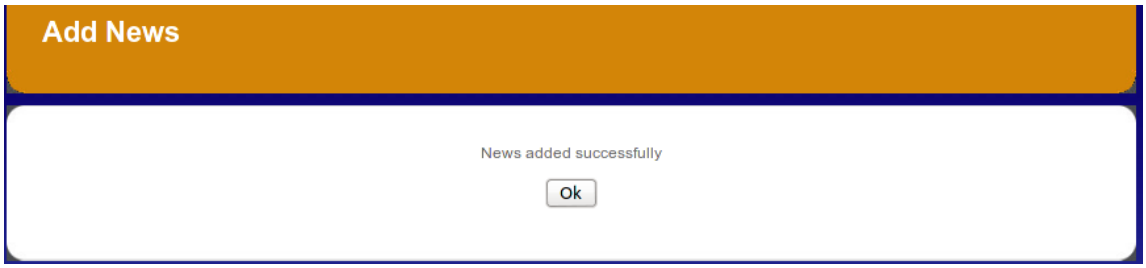
Actualización de un registro

```
$this->db->where('id', $id);
$this->db->update('mytable', $data);
Parámetros:
$id : id del registro a actualizar
mytable : nombre de la tabla
$object : un array con los datos a actulizar
```

9 – Retornaremos la cadena 'add' en caso de suceso o un mensaje de error al contrario al código Ajax.

10 – La respuesta 'add' hace que el código Ajax llame a la función c_news/endOperationAddNews() del controlador. Esta función se encarga de cargar la vista de fin de operación.

11 – Se fija la vista v_finalOperation fijando el mensaje de suceso del registro de la noticia.



La modificación y la supresión de una noticia está basada sobre el mismo esquema. Excepto que se hace un select al principio de la operación, a partir del modelo m_news, para seleccionar el registro que queremos modificar o suprimir. Con el mismo esquema para cada operación el mantenimiento del código se hace más fácil.

El reto para las aplicaciones web es la transmisión de datos entre el navegador y el servidor. El método clásico de diálogo utiliza mecanismos del World Wide Web, que son incorporados dentro todos los navegadores, y no necesita programación. Por el contrario, el funcionamiento de Ajax necesita de programar en JavaScript los intercambios entre el navegador y el servidor web. También se necesita de implementar las modificaciones que hay que efectuar en la página web a la recepción de las respuestas. Ajax nos da flexibilidad y control.

Transacciones

La clase database de Codeigniter, nos permite si se necesita aprovechar de la implementación de la gestión de las transacciones por la base de datos.

Con una base de datos MySQL debemos utilizar el mecanismo de almacenamiento de datos InnoDB en lugar de MyISAM.

Utilizar la gestión de transacción es muy sencillo con CodeIgniter. Entre las lineas de comando **$this->db->trans_start()** y **$this->db->trans_complete()** podemos tener varias peticiones hacia la base de datos. Si una petición genera un error un rollback se hará para mantener la integridad de la base de datos. Ejemplo en el modelo m_comments.php donde después la inserción de un comentario, actualizamos la tabla de las noticias. Las dos peticiones forman una sola transacción.

```
function insert_Comments($newComments)
{
        $this->db->trans_start();
   $res = $this->db->insert($this->table, $newComments);
        if (!$res) {
                // database config debug = false to work
                 $msg = $this->db->_error_message();
                 $num = $this->db->_error_number();
                 return  "Error(".$num.") ".$msg."\nContact web site administrator.";
        }
        $selectCol = 'nbr_comments';
        $this->db->select($selectCol);
        $records=$this->db->get_where('asy_news',array('id_news' =>
$newComments['id_news']));
        $data = $records->result_array();
        $dataGet = $data[0];
        $nbr_comments = $dataGet['nbr_comments'];
        $nbr_comments = $nbr_comments + 1;
        $data = array('nbr_comments' => $nbr_comments);
        $res = $this->db->update('asy_news', $data, 'id_news ='.$newComments['id_news']);
        if (!$res) {
                 $msg = $this->db->_error_message();
                 $num = $this->db->_error_number();
                 return  "Error(".$num.") ".$msg."\nContact web site administrator.";
        }
        $this->db->trans_complete();
        return 'add';
}
```

El Mapa Google maps

Para los datos de los proyectos quería fijar las oficinas de aduana en un mapa. Una librería basado sobre el API V3 de google maps me ayudó para este reto. Esta librería se puede encontrar en el enlace siguiente : http://biostall.com/codeigniter-google-maps-v3-api-library.

La librería se instala en la carpeta application/libraries. La integración en el código es muy sencillo. Primero se debe cargar la librería que se llama googlemaps, después se recupera los coordinaros de las oficinas desde la base de datos, se inicializa el objeto googlemaps y se crea el mapa.

```
$this->load->library('googlemaps');
$country = $this->m_country->get_CountryLongLat($id);
$country = $country[0];
$config['center'] = $country['longitude'].','.$country['latitude'];
$config['zoom'] = '7';
$this->googlemaps->initialize($config);
$this->set_info_office($id);
$data['map'] = $this->googlemaps->create_map();
```

3.2 Implantación

En este capítulo se presenta el pase a producción del sistema.

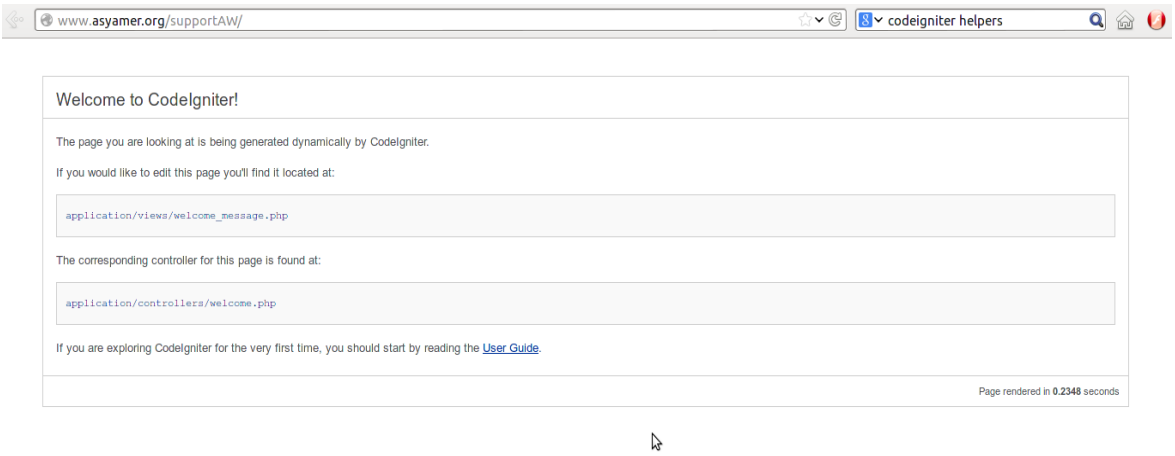
- **Instalación en entorno de producción :** es la implantación del sistema en ambiente de producción y la instalación de todo lo que es necesario para su funcionamiento.
- **Pruebas de implantación :** es la validación de la instalación del sistema en producción.
- **Nivel de servicio** : Se determinará según el nivel de servicio deseado, los recursos necesarios para cumplir los compromisos acordados. (soporte, horas de servicios...).

3.2.1 Instalación en entorno de producción

Inmotion, donde tenemos nuestro espacio para el proyecto, suministra una interfaz que permite a los administradores de instalar las aplicaciones o servicios que desean. Instalar una aplicación es muy sencillo. Debemos seleccionar el programa a instalar y arrancar la instalación. Después el script de instalación se encarga de guiarnos. Para la instalación de Codigniter, se nos pidió :

- la carpeta donde queremos instalar Codeigniter,
- el nombre de la base de datos que queremos aunque para Codeigniter so se necesita una base de datos para su instalación.

La verificación de la instalación de Codeigniter se hace conectándose al sitio en linea (www.asyamer.org/supportAW) y la página principal de Codeigniter debe fijarse.



El intérprete PHP, la herramienta PhPMyAdmin y el servidor Apache están instalados por defecto.

Ya tenemos ahora todo lo que se necesita para ahora instalar nuestra aplicación. La instalación tiene las etapas siguientes:

- generar el script de creación de la base de datos gracias a PhPMyAdmin de nuestro ambiente de desarrollo,
- utilizar PhpMyAdmin de producción para crear la base de datos y sus tablas con el script generado,
- cargar los datos iniciales,
- instalar los archivos de la aplicación
  - archivos de configuraciones,
  - controladores, modelos, vistas,
  - nuevas clases, helpers desarrollados para nuestra aplicación.

Ahora se necesita hacer las pruebas de implantación.

3.2.2 Pruebas de implantación

Las pruebas de implantación se apoyan sobre el plan de prueba definido en el apartado 2.2.3 Plan de Prueba.

En la medida que se instaló el sistema en un otro servidor es necesario repasar todos los escenarios de prueba para estar seguro que todo fue instalado correctamente. Estas pruebas necesitan una navegación completa a través de todo el sitio.

Desde las primeras pruebas podremos apreciar si los servicios ofrecidos por el servidor de producción son satisfactorios sobre todo con lo relacionado al tiempo de respuesta.

Según los resultados de las pruebas y la apreciación de los servicios ofrecidos por el servidor podremos tomar decisiones si se necesita hacer modificaciones y/o mejoras implicando nuevas pruebas completas o parciales antes de abrir completamente el sitio al público.

3.2.3 Nivel de servicios

El nivel de servicio se comparte en dos entidades:

- Nivel de servicios del propio servidor : para esta parte dependemos de los servicios ofrecidos por el suministrador de alojamiento la empresa Inmotion. Inmotion nos garantiza una accesibilidad y un funcionamiento del servidor sin interrupción 24/7/365, un soporte técnico, herramientas e interfaces para la administración básica del servidor.
- Nivel de servicio del sitio web : es primordial alimentar al sitio web debido a que es una herramienta de comunicación. Hay que determinar quién estará encargado de alimentar el sitio con informaciones.
    - un administrador general del sitio basado en el Centro Regional de las Américas a Caracas,
    - un administrador de las descargas ubicado en el equipo central de Ginebra.
    - dos administradores de noticias, uno basado en Ginebra del equipo central y uno en el Centro Regional de las Américas en Caracas,
    - cada país tendrá un administrador de los datos de su proyecto.
    - 24h para dar una respuesta por parte del administrador una vez reportado y encontrado un problema.
    - Los horarios de servicio serán los de apertura de las oficinas.

Las tareas de mantenimiento por parte del administrador general serán:

- Mantener al día de las tablas de referencia : versión de java, de soclass, de SIDUNEA World, de sistema operativo y base de datos.
- Hacer un respaldo semana del contenido de la base de datos.
- Hacer un respaldo de la carpeta upload donde se ubicarán los archivos descargables : documentaciones, versiones, archivos de nuevas funcionalidades de los proyectos.

4 CONCLUSIONES

La idea de este proyecto se originó de una reflexión. Aunque el programa SIDUNEA World sea el más utilizado en el mundo ya que es desplegado en más de 100 países que lo adoptaron para la gestión de su administración aduanera, siempre nos ha sido difícil compartir nuestros conocimientos a nivel técnico. Si usted visita diferentes países que utilizan SIDUNEA World, comprobará que cada uno de ellos ha desarrollado nuevas funcionalidades sin haber compartido estos avances y/o logros con demás proyectos.

El objetivo de este sitio web es el de empezar un intercambio entre los proyectos sobre las nuevas funcionalidades desarrolladas con el fin de continuar y mejorar SIDUNEA World. La falta de difusión de informaciones técnicas, de documentación es también una debilidad de nuestra parte que contrasta con el éxito del proyecto al nivel mundial.

Al haber trabajado, desde hace muchos años en el mundo de la tecnología cliente-servidor, este proyecto de fin de estudio me dio la oportunidad de ir a visitar un nuevo universo, el de la web. Tras varios meses de trabajo duro llega el resultado final, mi primer sitio web tiene vida. La satisfacción del trabajo finalizado es gratificante pero por otra parte puedo identificare al mismo tiempo muchos puntos a mejorar. Debido a mi falta de experiencia con el intérprete PHP, las hojas de estilo, Javascript, y Ajax hay cosas que no se han echo de la mejor manera desde el principio por desconocimiento del entorno.

Sin embargo, estoy feliz de llevar a cabo este proyecto considerando que es una primera versión que pronto tendrá varias mejoras y extensiones. Espero también que sea satisfactorio para los usuarios y que todos comencemos a compartir nuestras experiencias.

5 BIBLIOGRAFÍA

Página principal del Servidor Apache[online]. URL: http://www.apache.org/

Página principal Codeiigniter [online]. URL: http://ellislab.com/codeigniter

Página foro Codeigniter [online]. URL: http://ellislab.com/forums/

Página principal de los estándares de la web : http://www.w3.org/

Página principal de JQuery [online]. URL: http://jquery.com/

Página principal de MySQL [online]. URL: http://www.mysql.com/

Página oficial de PHP [online]. URL: http://www.php.net

Página principal w3schools [online]. URL: http://www.w3schools.com/

Página hoja de estilo [online]. URL: http://www.w3.org/Style/CSS/

Página Google maps API V3 [online]. URL: https://developers.google.com/maps/documentation/javascript/

Página principal servidor SVN [online]. URL: http://subversion.tigris.org/

Página principal entorno de desarrollo Aptana [online]. URL: http://www.aptana.com/