

UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO

# **“SISTEMA WEB INTEGRADO DE PLANIFICACIÓN PEDAGÓGICA Y CONTROL DE PAGOS”**

*INSTITUTO EXCLUSIVO DE IDIOMAS – LA PAZ*

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA MENCIÓN:  
INGENIERÍA DE SISTEMAS INFORMÁTICOS

AUTOR: CARLOS ALBERTO RAMOS KANTUTA

TUTOR: LIC. EFRAÍN SILVA SÁNCHEZ

REVISORA: LIC. CELIA ELENA TARQUINO PERALTA

LA PAZ – BOLIVIA

2007

*Dedicado a:*

*Mis padres Estaban y Julia a quienes con su esfuerzo y ayuda incondicional, me dieron la fuerza para lograr mis metas.*

*La vida por darme otra oportunidad.*

## AGRADECIMIENTOS

A mi familia por estar presente en todo momento, compartiendo en tiempos de alegrías como en tristezas, por ese apoyo incondicional que es una tónica para afrontar el día a día, simplemente decirles ¡¡¡GRACIAS!!!

A Dios por darme la vida.

Al licenciado Efraín Silva Sánchez por su ayuda incondicional, por brindarme consejos cuando lo necesitaba, y el apoyo hasta la culminación del presente proyecto.

A la licenciada Celia Elena Tarquino Peralta, por su ayuda, consejos y brindar sus observaciones del proyecto, que en su momento me permitieron tomar el rumbo cuando me descarrilaba.

A la directora del Instituto Exclusivo de Idiomas Anne Marie Van Swinderen, por su confianza y colaboración desde un principio para desarrollar el presente proyecto.

Al personal del Instituto, en especial a Teresa y Iris que me brindó su ayuda y colaboración para el proyecto, gracias.

A mis amigos Marcelo, Juan Carlos y Ramiro por brindar su amistad.

Además brindar un extenso gracias a todas las personas que me acompañaron en toda la vida universitaria, a los docentes que me brindaron cada uno su propio punto de vista al momento de enseñar, a mis amigos y conocidos que hicieron más llevadera la vida universitaria.

Al Dña. María y el personal administrativo que colaboro con el papeleo, admiro su trabajo.

**GRACIAS.**

## RESUMEN

Desde el año 2005 se empieza a propagar el concepto de web 2.0, promulgando una red mas social o humana y acorde a la tecnología actual, concepto clave para el éxito de aplicaciones web, se quiere abordar la misma de manera mas practica dentro del proyecto.

El instituto esclusico de Idiomas, es un instituto que brinda la enseñanza de idiomas, primordialmente el español, a turistas, becarios y principalmente a personas del mundo a través del internet, a través de videoconferencias u aulas virtuales.

El presente proyecto trata de solucionar el creciente problema de planificación pedagógica dentro del Instituto, manejar una planificación para cada curso, asignar y permitir reservar una actividad pedagógica, permitir elaborar un horario a estudiantes y profesores y a partir de sus actividades registradas elaborar un control de deudas a estudiantes y calculo de comisiones a profesores, además se requiere un registro de traducciones y sus pagos que se elaboran dentro del instituto.

Dentro de la aplicación se investigara y aplicara el desarrollo de aplicaciones web bajo el modelo y patrón de desarrollo de software MVC (Modelo-Vista-Controlador), el cual facilita el desarrollo de aplicaciones web, separando la capa de lógica de negocios con la capa de presentación. Además para la gestión del proyecto se llevara a cabo bajo la mitología SCRUM, considerado como una metodología ágil de trabajo. La implementación se utilizara el lenguaje de programación Ruby y la plataforma para el desarrollo de aplicaciones web: Rails, que juntos brindan una robusta plataforma de desarrollo ágil, novedoso y actual.

También se aplicara el concepto de sitios web 2.0, aplicando la tecnología Ajax no obstrusivo y el manejo de recursos web REST.

# Capítulo I

## 1.1 INTRODUCCIÓN

La necesidad actual de sistemas de accesibles, integrados, ubicuos<sup>1</sup> y de fácil manejo es la demanda actual dentro de las empresas e instituciones, una solución a esta demanda es el desarrollo de sistemas Web, ya que para su manejo solo se requiere de un navegador web por parte del cliente.

Gracias a la creación de la World Wide Web, la red de redes, supuso una explosión en la difusión de la información, pero siguiendo una evolución natural ahora se vuelca a un nivel más social con lo que vendría a ser la Web 2.0, entendiendo esta visión se han creado sitios web que satisfagan estas necesidades de tipo social como ser blogs<sup>2</sup>, buscadores personales, marcadores de información y las redes sociales, además dentro del concepto de web 2.0, se integra la tecnología llamada Ajax<sup>3</sup> que primordialmente reduce los tiempos de respuesta entre la aplicación web y el usuario, logrando una mejor experiencia al visitar la página y posibilitando su vuelta al sitio web. [FUMERO-ROCA, 2007]

El Instituto Exclusivo de Idiomas, en su labor de enseñanza propone la enseñanza del español on-line, que utiliza como principal agente al internet ofreciendo clases para personas de todo el mundo, esto como un resultado a la demanda requerida de la web, pero de una manera más amena y social como así se propone en lo que engloba el concepto de Web 2.0, por tanto se requiere de un sistema de planificación de actividades pedagógicas que facilite la coordinación de diferentes actividades entre profesores y estudiantes.

---

<sup>1</sup> Que está presente en todas partes a un mismo tiempo.

<sup>2</sup> Sitio web donde las personas publican sus diarios personales narrando sobre sus pasatiempos o experiencias; entradas online adaptadas cronológicamente de conexiones con otros sitios y tópicos que contienen observaciones personales de los usuarios, weblog. [Wikipedia, 2007]

<sup>3</sup> Asynchronous Java Script and XML, son llamadas asincrónicas desde el navegador web al servidor, lo cual da el efecto de respuesta inmediata dentro de la página web

El presente proyecto trata de solucionar el creciente problema de planificación pedagógica dentro del Instituto, esto es que a cada profesor por clase debe de tener asignado estudiantes y un aula, pero se encuentran frecuentemente con problemas de choque de horarios, falta de aulas, o posibles cancelaciones de la clase repentinos por parte de estudiantes y profesores, lo que dificulta el proceso de enseñanza del estudiantes y la inconformidad de profesores y la dirección, además para poder tomar clases los estudiantes deben de realizar el pago por los servicios de enseñanza del idioma español, este cobro se realiza presencialmente o través del sistema de cobros en-línea PayPal<sup>4</sup>, una vez realizado el cobro por cualquiera de los métodos anteriormente descritos se asigna crédito al estudiante, con este crédito puede realizar su registro de clases con algún profesor y otras actividades secundarias descontando de su crédito cada vez que registra una clase u otra actividad, y que a partir del registro de estas clases ya pasadas se procede al cálculo y pago a los profesores.

## 1.2 ANTECEDENTES

### 1.2.1 DE LA INSTITUCIÓN

#### 1.2.1.1 DESCRIPCIÓN DE LA INSTITUCIÓN: INSTITUTO EXCLUSIVO DE IDIOMAS

El instituto Exclusivo de Idiomas, es creado a partir de la necesidad de muchos extranjeros que llegan al país o cualquier parte de América latina, y se encuentran en la necesidad de hablar el idioma español y conocer las costumbres locales de manera que se puedan desenvolver de una manera más normal, mejorando su experiencia personal, posibilitando así una residencia permanente.

El Instituto Exclusivo de idiomas i.e. es una organización de estructura familiar, especializada en ofrecer enseñanza de calidad de los idiomas español y aimara como segundo idioma, a clientes extranjeros y bolivianos.

---

<sup>4</sup> Servicio de cobros a través de internet

El Instituto Exclusivo de idiomas i.e. es una organización joven e innovadora, la cual entra al mercado nacional en Enero de 2006, bajo la dirección de la Sra. Anne Marie Van Swinderen; quien crea una organización con una visión innovadora de la enseñanza que se basa en el servir y no servirse del cliente.

#### 1.2.1.2 MARCO GLOBAL

Rubro principal del negocio: Educación del idioma español como segunda lengua.

**Misión:** Desarrollar un instituto de enseñanza del idioma español de calidad, creando diferentes oportunidades en la forma en la que el individuo puede aprender este idioma.

**Visión:** Atraer individuos en busca del aprendizaje del idioma español, ofreciéndoles un producto de calidad.

**Mega:** Enseñanza uno a uno vía Internet. De esta manera, la empresa puede explotar las riquezas intelectuales de Bolivia sin causar cansancio mental o abusar del sistema salarial bajo local.

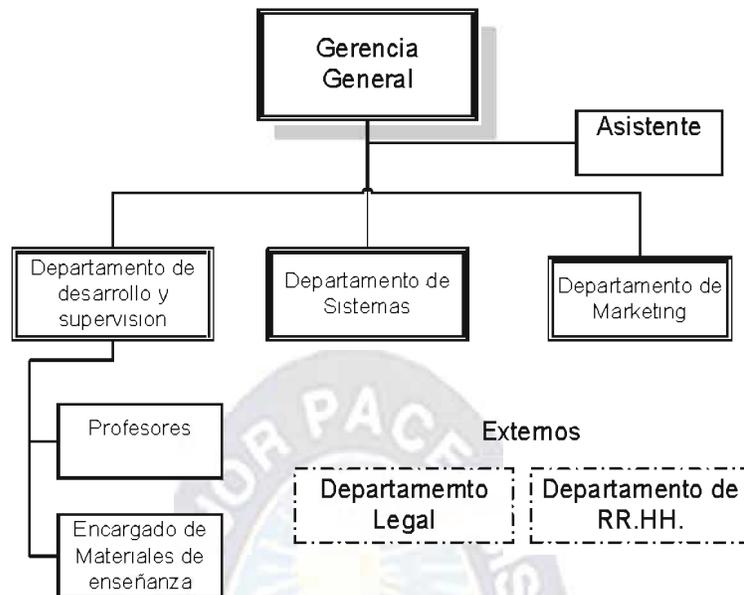
#### 1.2.1.3 ESTRUCTURA ORGANICA DE INSTITUTO EXCLUSIVO

La organización tiene una estructura horizontal con liderazgo democrático. La figura 1.1 muestra el organigrama dentro de la institución.

##### I) RECURSOS HUMANOS

Actualmente dentro del instituto trabajan alrededor de 18 personas.

##### II) RECURSOS TÉCNICOS



**FIGURA 1.1** ORGANIGRAMA DEL INSTITUTO EXCLUSIVO DE IDIOMAS

**Fuente:** Instituto Exclusivo de Idiomas

Para poder alojar la aplicación web la directora del Instituto Exclusivo de Idiomas, recurre a los servicios ofrecidos por la empresa Norteamericana RailsPlayground, que brindan los servicios de Servidores Privados Virtuales (VPS<sup>5</sup>), con soporte a base de datos MySQL y lenguajes de programación PHP, Ruby y la posibilidad de compra de dominios propios. Se usará este servicio para la puesta en marcha de la aplicación web.

#### 1.2.1.4 SITUACIÓN ACTUAL

El Instituto Exclusivo de Idiomas está legalmente establecido, con resolución ministerial Nro. 248/07, brinda los servicios de enseñanza a adultos de los idiomas: español, inglés y aimara.

#### 1.2.2 DEL PROYECTO

Revisando trabajos similares anteriores en la biblioteca de Informática, se encontraron los siguientes proyectos, que sirven de guía para la realización del presente proyecto:

<sup>5</sup> Virtual Private Server, emulación mediante software de un servidor, esto reduce drásticamente los costos de instalación y mantenimiento, porque dentro de un servidor pueden existir más de un VPS.

SISTEMA DE ADMINISTRACIÓN Y CONTROL DE PASAJES AÉREOS VÍA WEB, Sistema Web para la administración y control de asignación de pasajes aéreos dentro del Senado Nacional Boliviano, diseñado bajo la metodología de diseño navegacional OOHDM.

ASIGNACIÓN DE RECURSOS DE INFRAESTRUCTURA ACADÉMICA CON ALGORITMOS GENÉTICOS, Tesis de Grado en el cual se genera un algoritmo que define una eficiente asignación de recursos como ser: profesores o docentes, aulas y horarios. Maximizando el uso de estos a partir del uso de algoritmos genéticos, se toma como pauta por los requerimientos del proyecto [MORALES, 2000].

SISTEMA ADMINISTRATIVO Y ACADÉMICO "EDUCA", Sistema integrando para el colegio María Auxiliadora desarrollando módulos para las partes administrativo y académico para el uso de Director, profesores, alumnado y padres de familia, controla el pago de pensiones mensuales y la gestión académica [LAURA, 2004].

APLICACIÓN WEB DE INFORMÁTICA EMPLEANDO USABILIDAD, Proyecto de Grado donde se desarrolla un sitio web para el uso de docentes y universitarios de la carrera de Informática, se definen conceptos de usabilidad para el desarrollo de sitios web.

## 1.3 ANÁLISIS Y PLANTEAMIENTO DEL PROBLEMA

### 1.3.1 ANÁLISIS DE LA PROBLEMÁTICA

El Instituto Exclusivo de Idiomas promueve la enseñanza del español a personas del mundo que deseen aprenderlo, es en ese sentido que el Instituto Exclusivo de Idiomas enseña a personas extranjeras el idioma español dentro de sus instalaciones, además se dictan clases virtuales el cual consiste en tener un sitio web únicamente para promocionar las clases por internet, el cliente visita el sitio, si le convence realiza un pago a una cuenta en el extranjero, coordina un horario con un profesor y empieza con las clases virtuales.

En un principio (un año atrás), era fácil el control y la administración del total de clases que se realizan dentro del instituto, registrados dentro de planillas elaboradas en Excel y luego

impresas, con el pasar del tiempo el Instituto Exclusivo de Idiomas a crecido hasta llegar actualmente a 15 profesores, 65 alumnos y esto tiende a aumentar. Además la administración precisa de elaborar planillas de pago a través de las horas que ha trabajado el profesor, también se requiere el poder ver la situación financiera de cada estudiante para poder figurarlo entre los que puedan tomar clases, y si tienen alguna deuda el poder bloquearlos hasta que normalicen sus cuentas.

Es por eso que se deben de responder a las siguientes necesidades dentro del Instituto:

- Registro y clasificación manual de estudiantes, profesores y traductores, No se tiene un actual y correcto registro de estudiantes, profesores y traductores, lo que dificulta la gestión de usuarios dentro del instituto.
- El proceso actual de inscripción y elección de clases es manual, lo cual impide una rápida consulta de que actividades registradas tienen los estudiantes para el día o semana.
- Perdida de recursos y tiempo en la elaboración de la planificación para el día o semana por parte de profesores.
- Manejo inadecuado de recursos comunicacionales. El Instituto tiene acceso completo a internet, pero no es aprovechado al máximo ya que se utilizan más el teléfono o esperar que se encuentren presencialmente los profesores, dificultando la comunicación dentro del Instituto.
- Actualmente los pagos de estudiantes se registran bajo recibos lo que implica que no se pueda saber oportunamente si el estudiante tiene crédito además dificulta en el cálculo de deudas contraídas por clases con el instituto.
- Los profesores registran manualmente en hojas sus clases pasadas que están sujetas a perderse o las anotaciones no son confiables, lo que dificulta el cálculo de comisiones y pago a profesores.
- Además, de enseñar idiomas el Instituto realiza traducciones, los cuales son registrados manualmente en una hoja de Excel, se necesita recepcionar los documentos a traducir vía web y elaborar un registro de esta.

- No se cuenta con informes estadísticos de estudiantes, clases, ingresos y egresos para la toma de decisiones.

### 1.3.2 PLANTEAMIENTO DEL PROBLEMA

El instituto exclusivo con sede en La Paz-Bolivia y con presencia internacional tiene como función la enseñanza del español y debe de hacer un seguimiento de clases de sus alumnos, sus pagos, reservas de horarios, pagos a los profesores y coordinar los horarios de clases entre profesores, alumnos on-line, por lo que se formula la siguiente cuestión:

*¿El Sistema Web integrado de planificación pedagógica y control de pagos, mejorara el proceso de registro de usuarios, gestión y la planificación de actividades pedagógicas y la gestión de traducciones, así como el control de pagos de estudiantes y profesores logrando un uso eficaz tanto de tiempo y empleo de recursos pedagógicos y materiales?*

## 1.4 OBJETIVOS

### 1.4.1 OBJETIVOS GENERALES

El desarrollar e implementar el sistema web integrado de planificación pedagógica y control de pagos, que permitirá el registro de usuarios, planificación de actividades pedagógicas, manejo de traducciones y el control de pagos de estudiantes, profesores y traductores dentro del Instituto Exclusivo de Idiomas aplicando el concepto de web 2.0.

### 1.4.2 OBJETIVOS ESPECÍFICOS

- Registrar nuevos usuarios clasificándolos y asignando roles automáticamente.
- Inscripción de estudiantes a los cursos de acuerdo a formato del instituto y permitir que estudiantes puedan reservar y ver sus actividades pedagógicas a través del sistema Web.

- Generar los horarios por estudiantes, profesores o aulas automáticamente por día o semana, permitiendo agilizar la planificación de recursos.
- Permitir a estudiantes, profesores y traductores envíos de notificaciones a través de correo electrónico y chat.
- Gestionar y elaborar reportes sobre el crédito y clases pasadas de estudiantes.
- Calcular y generar las comisiones de pago a profesores a partir de sus actividades registradas dentro del sistema.
- Elaborar un modulo de registro de traducciones y sus pagos.
- Generar estadísticas tomando criterios de fechas y registros del sistema que ayudaran en toma de decisiones.

## 1.5 JUSTIFICACIONES

Se plantea un sistema de información debido a la necesidad imperiosa de información integra, precisa, oportuna y segura tanto para los alumnos, profesores y la administración.

### 1.5.1 JUSTIFICACIÓN TÉCNICA-CIENTÍFICA

El proyecto al implementarse se justificara técnicamente por que representara un salto tecnológico e innovación de procesos automatizados dentro del Instituto.

Además se avanzara a un nuevo enfoque al desarrollo ágil de software, dentro de la institución al implantarse agilizara la administración de las actividades de los profesores, empleando las nuevas tecnologías en el área informática, y gracias a la característica del desarrollo permitirá la comunicación interactiva dentro de los niveles de acceso de los usuarios.

El Instituto actualmente posee para el alojamiento web servidores alquilados del extranjero (Estados Unidos), los cuales se deben de configurar remotamente para la implementación, estos soportan todos los requerimientos técnicos que necesita el proyecto, dentro del

Instituto ya cuenta con una red interna en las que están conectadas 9 equipos, las características de hardware son :

- Procesador Pentium 4.
- Memoria Ram de 512 Mb.
- Disco Duro de 80 Gb.

Instaladas en cada equipo el sistema operativo Windows XP, utilizando los navegadores para internet: MS Internet Explorer 6 y Mozilla Firefox 2.0, con estas características técnica de hardware y software, es posible implementar el sistema planteado.

### 1.5.2 JUSTIFICACIÓN ECONÓMICA

La puesta en marcha del sistema permitirá mejorar el rendimiento de los horarios de enseñanza por los profesores y el tiempo de procesos administrativos con los que cuenta el Instituto, además de los costos de impresión y papelería.

En su implementación la aplicación web podrá colaborar en la agilización y sincronización de las distintas actividades de los profesores optimizando y mejorando el uso de recursos dentro de la institución, esto por el reducir los tiempos y costos del seguimiento de actividades.

El costo del desarrollo del sistema será ínfimo, a comparación del mercado normal y para la puesta en marcha no se requerirá de inversión ya que se tiene un servidor web alquilado y el dominio .com correspondiente.

### 1.5.3 JUSTIFICACIÓN SOCIAL

La ejecución del sistema mejorara la comunicación y la interacción entre los estudiantes-profesores y profesores-administración

El impulsar la nueva generación de personas que se encuentran familiarizados con el uso de la tecnología, y permitir ayudar a las personas que todavía no aprovechan al máximo de las tecnologías, minimizando el esfuerzo con la eliminación de procesos repetitivos.

## 1.6 MÉTODOS Y TÉCNICAS

Los métodos a utilizar son:

La gestión del proyecto bajo SCRUM<sup>6</sup> una reciente metodología ágil de desarrollo, apoyado por algunos artefactos del UML.

La arquitectura de la aplicación se basara en el patrón arquitectónico MVC<sup>7</sup>, que indica pasos para poder separar la lógica de negocios, con la presentación de datos. Además se utiliza plataformas de desarrollo conocidas como frameworks, en este caso el framework Ruby on Rails.

La arquitectura del servicio, se utilizara el concepto REST<sup>8</sup> que propugna que en total el sitio no describa acciones, sino recursos.

La técnica es de introducir Ajax, para poder tener una interacción dinámica y rápida con el usuario, y además es una característica para describir un sitio web 2.0.

Además se utilizaran herramientas Open source<sup>9</sup> que reducen los costos y que nos permiten trabajar profesionalmente y que serán descritos en el siguiente capítulo.

## 1.7 ALCANCES Y APORTES

### 1.7.1 ALCANCES

El Sistema Web de planificación de actividades didácticas llevara a cabo principalmente estas tareas:

- Realizar la inscripción de nuevos usuarios.

---

<sup>6</sup> Metodología de gestión de desarrollo ágil de software.

<sup>7</sup> Patrón de arquitectura para la creación de software.

<sup>8</sup> Patrón de diseño del servicio, permite acceder al recurso web fácilmente por que es auto descriptivo.

<sup>9</sup> Código abierto, es software que se distribuye con su código fuente y es gratuito.

- Cada profesor podrá anotar, editar y borrar sus horarios disponibles.
- Cada estudiante podrá registrar, modificar y borrar, de acuerdo a reglas dentro del instituto, sus clases.
- Manejar una planificación por curso dictado y está disponible para los estudiantes inscritos en el curso.
- Mostrar los horarios por día y semana.
- Los administradores manejan plenamente los horarios de profesores y estudiantes.
- Registran los pagos de estudiantes.
- Calcular y generar los pagos de profesores y traductores.
- Manejo de recursos del instituto: horarios disponibles, aulas, crédito de estudiantes.
- Asignación, modificación y eliminación de roles de usuario dentro del sistema.
- Envío de emails a cualquier usuario, notificando cualquier evento.

### 1.7.2 APORTES

Dentro de la aplicación se investigara y aplicara el desarrollo de aplicaciones web bajo el modelo y patrón de desarrollo de software MVC (Modelo-Vista-Controlador), el cual facilita el desarrollo de aplicaciones web, separando la capa de lógica de negocios con la capa de presentación. Además para la gestión del proyecto se llevara a cabo bajo la metodología SCRUM. La implementación se utilizara el lenguaje de programación Ruby y la plataforma para el desarrollo de aplicaciones web: Rails, que juntos brindan una robusta plataforma de desarrollo profesional, ágil, novedoso y actual.

También se adentrara en concepto de sitios web 2.0, aplicando la tecnología Ajax no obstrusivo que se bien es una mejora, su abuso lograría confundir o tener una experiencia desalentadora para el usuario final.

Además se realizara la recopilación de anteriores experiencias en otras aplicaciones web y la aplicación de buenas prácticas de programación, investigación de estándares web, y manejo de hojas de estilo (CSS) que ayudaran a la usabilidad del Sistema Web.

# Capítulo II

En este capítulo se definen los fundamentos teóricos y metodológicos bajo los cuales se desarrollará el proyecto, primero describiremos la metodología, posteriormente la arquitectura, la técnica, y terminando la usabilidad, seguridad y calidad del sistema.

## 2.1 EL PROCESO UNIFICADO

El proceso unificado de Rational (RUP, Rational unified process) es un proceso para el desarrollo de un proyecto de un software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto. Como 3 características esenciales está dirigido por los Casos de Uso: que orientan el proyecto a la importancia para el usuario y lo que este quiere, está centrado en la arquitectura: que Relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y es iterativo e incremental: donde divide el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

### 2.1.1 PRINCIPIOS

Como filosofía RUP maneja 6 principios clave [WIK07]:

#### a) Adaptación del proceso

El proceso deberá adaptarse a las características propias de la organización. El tamaño del mismo, así como las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto.

#### b) Balancear prioridades

Los requerimientos de los diversos inversores pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos.

### **c) Colaboración entre equipos**

El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes, resultados, etc.

### **d) Demostrar valor iterativamente**

Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados.

### **e) Elevar el nivel de abstracción**

Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software o esquemas (frameworks<sup>10</sup>) por nombrar algunos. Éstos se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con UML<sup>11</sup>.

### **f) Enfocarse en la calidad**

El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción.

## **2.1.2 CARACTERÍSTICAS**

### **2.1.2.1 PROCESO DIRIGIDO POR CASOS DE USO**

Un sistema software ve la luz para dar servicio a sus usuarios, normalmente un sistema tiene muchos tipos de usuarios, cada tipo de usuario se representa por un actor. Los actores utilizan el sistema interactuando con los casos de uso (Secuencia de acciones que el sistema

---

<sup>10</sup> Plataforma de trabajo, conjuntos y/o bibliotecas de funciones comunes organizadas, para el desarrollo de software.

<sup>11</sup> Unified Model Language, Lenguaje unificado de modelado, lenguaje orientado a objetos, para describir actores y funciones.

lleva a cabo para ofrecer algún resultado), que representan los requisitos funcionales del sistema.

Los casos de uso, no solo especifican los requisitos de un sistema sino guían su diseño, implementación y pruebas.

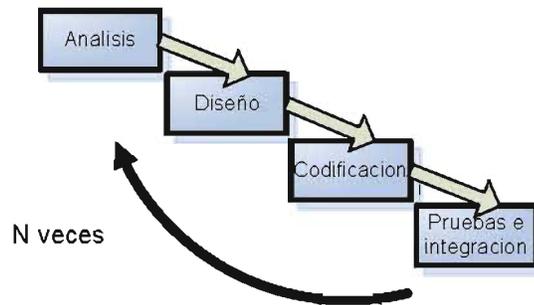
### 2.1.2.2 PROCESO CENTRADO EN LA ARQUITECTURA

La arquitectura involucra los elementos más significativos del sistema y esta influenciada entre otros por plataformas de software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Los casos de uso guían el desarrollo de la arquitectura se realimenta en los casos de uso, los dos juntos permiten conceptualizar, gestionar y desarrollar adecuadamente el software.

Gracias a las características del modelado se han creado diferentes patrones de arquitectura, dentro del presente proyecto se manejará el patrón de arquitectura de software: **Modelo Vista Controlador**, descrito en la sección 2.3.

### 2.1.2.3 PROCESO ITERATIVO E INCREMENTAL

Para hacer más manejable un proyecto se recomienda dividirlo en mini proyectos, siendo cada uno de ellos una iteración. Cada iteración tiene un ciclo de vida en cascada a menor escala. El ciclo de vida iterativo se basa en la evolución de prototipos ejecutables que se muestran a los usuarios y clientes. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. Las actividades se encadenan de una mini cascada con un alcance para los objetivos de la iteración. (Figura 2.1)



**FIGURA 2.1** MINI CASCADA DENTRO DE CADA ITERACION

**FUENTE:** THE UNIFIED DEVELOPMENT PROCESS. JACOBSON, BOOCH AND RUMBAUGH ADDISON- WESLEY, 1999

### 2.1.3 ESTRUCTURA DEL PROCESO

El proceso puede ser descrito en dos dimensiones o ejes, estos son:

#### EJE HORIZONTAL

Representa el tiempo y es considerado el eje de los aspectos dinámicos del proceso. Indica las características del ciclo de vida del proceso expresado en términos de fases, iteraciones e hitos. Se puede observar en la Figura que RUP consta de cuatro fases: Inicio, elaboración, construcción y Transición.

Como se mencionó anteriormente cada fase se subdivide a la vez en iteraciones

#### EJE VERTICAL

Representa los aspectos estáticos del proceso. Describe el proceso en términos de componentes de proceso, disciplinas, flujos de trabajo, actividades, artefactos y roles (Figura 2.2).

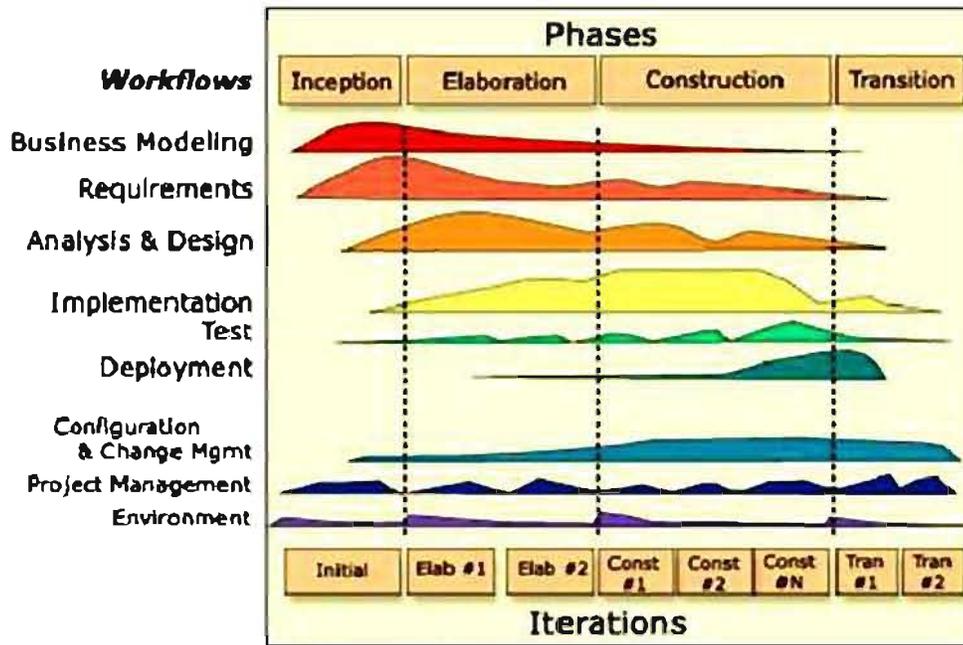


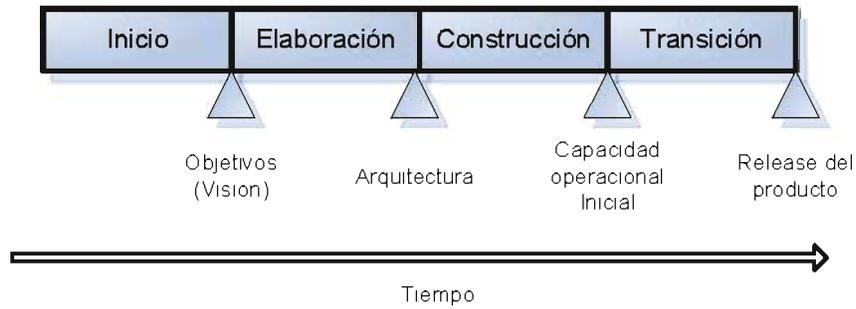
FIGURA 2.2 FASES DEL DESARROLLO RUP

FUENTE: THE UNIFIED DEVELOPMENT PROCESS. JACOBSON, BOOCH AND RUMBAUGH ADDISON- WESLEY, 1999

#### 2.1.4 ESTRUCTURA DINÁMICA DEL PROCESO, FASES E ITERACIONES

RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un producto. Cada ciclo concluye con una generación de producto para los clientes. Cada ciclo consta de cuatro fases: Inicio elaboración Construcción y transición. Cada fase se subdivide a la vez en iteraciones, el número de iteraciones en cada fase es variable.

Cada fase se concluye con un hito bien definido, un punto en el tiempo en el cual se deben tomar ciertas decisiones críticas y alcanzar las metas clave antes de pasar a la siguiente fase, ese hito principal de cada fase se compone de hitos menores que podrían ser los criterios aplicables a cada iteración. Las fases y sus respectivos hitos se ilustran en la figura 2.3:



**FIGURA 2.3 FASES E HITOS EN RUP**

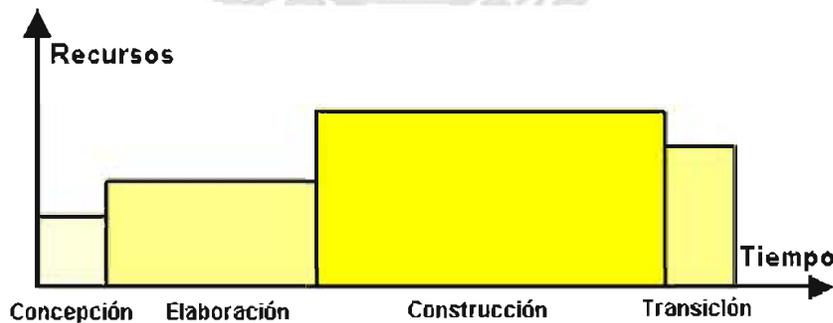
**FUENTE:** PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE, 1999

La duración y esfuerzo dedicado en cada fase es variable dependiendo de las características del proyecto. Sin embargo, la tabla 2.1 ilustra porcentajes frecuentes al respecto. Consecuente con el esfuerzo señalado, la figura 2.4 ilustra una distribución típica de recursos humanos necesarios a lo largo del proyecto.

	Inicio	Elaboración	Construcción	Transición
<b>Esfuerzo</b>	5%	20%	65%	10%
<b>Tiempo Dedicado</b>	10%	30%	50%	10%

**TABLA 2.1 DISTRIBUCION TIPICA DE ESFUERZO Y TIEMPO**

**FUENTE:** PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE, 1999



**FIGURA 2.4 DISTRIBUCION TIPICA DE RECURSOS HUMANOS**

**FUENTE:** PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE, 1999

### 2.1.5 CICLOS Y FASES

RUP divide el proceso de desarrollo en ciclos, teniendo un producto al final de cada ciclo.

Cada ciclo se divide en cuatro fases:

### 2.1.5.1 INICIO

Su meta principal es lograr el consenso de todos los involucrados acerca de los objetivos del ciclo de vida del proyecto. Es muy importante especialmente en proyectos nuevos en que existen riesgos significativos en el negocio o la implementación de los requisitos y deben ser solucionados para que el proyecto proceda. Para los proyectos que se enfocan en mejorar sistemas existentes, esta fase es breve, pero aun así centrada en asegurar que el proyecto vale la pena y se puede realizar. Además establece los objetivos para el ciclo del vida del producto. En esta fase se establece el caso del negocio con el fin de delimitar el alcance del sistema, saber que se cubrirá y delimitar el alcance del proyecto.

**HITO.** Establecer el ámbito del producto, la identificación de los principales riesgos y la viabilidad del proyecto.

### 2.1.5.2 ELABORACIÓN

El propósito de la etapa de elaboración es crear la línea base de la arquitectura para el ciclo de vida del producto para así disponer de unos cimientos sólidos sobre los que se basara el gruesos del esfuerzo de diseño e implantación durante la fase de construcción. La arquitectura evoluciona de los requisitos más significativos considerados (Aquellas que tiene un fuerte impacto en la arquitectura del sistema) y la evaluación de riesgos.

La estabilidad de la arquitectura se avalúa mediante el uso de prototipos de arquitectura. En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieren con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable el caso del negocio.

**HITO.** Obtener una línea base de la arquitectura del sistema, capturar la mayoría de los requisitos y reducir los riesgos principales así como permitir la escalabilidad del equipo del proyecto durante la fase de construcción.

### 2.1.5.3 CONSTRUCCIÓN

En la fase de construcción de deben de aclarar los requisitos restantes y completar el desarrollo del sistema basándose en la arquitectura que se ha añadido a la línea base. Puede ser vista como un proceso de fabricación donde se hace énfasis en la administración de los recursos y el control de operaciones para optimizar costos planificaciones y calidad. En este sentido la administración experimenta una transición del desarrollo de propiedad intelectual durante las fases de inicio y elaboración al desarrollo de productos instalables durante la Construcción y transición.

Su capacidad principal es alcanzar la capacidad operacional del producto. En esta fase a través de las sucesivas iteraciones e incrementos se desarrolla un producto software, listo para operar, este es frecuentemente llamado versión beta<sup>12</sup>.

**HITO.** Desarrollo del sistema con calidad de producción y prepararse para la entrega al equipo de transición. Toda la funcionalidad debe haber sido implementada y las pruebas para el estado beta de la aplicación completadas. Si el proyecto no logra alcanzar este hito, entonces la transición deberá posponerse una iteración.

### 2.1.5.4 TRANSICIÓN

La atención se enfoca en asegurar que el software está disponible para los usuarios finales, realizar la entrega del producto. Incluye las pruebas del producto como parte de su preparación para ser entregado y además la realización de ajustes menores en respuesta a la retroalimentación recibida de los usuarios.

**HITO.** Consiste en decidir si los objetivos se cumplieron y si debe comenzarse otro ciclo de desarrollo. Es el resultado de la revisión y la aceptación por parte del cliente de los artefactos que le han sido entregados.

---

<sup>12</sup> La última fase del examen de un programa software antes de salir al mercado.

## 2.2 WEB 2.0

La Web 2.0 define un cambio del modo estructural y social de lo que tradicionalmente se conoce al internet, es una fusión de lo social (cómo y dónde nos comunicamos y relacionamos) y lo tecnológico (nuevas herramientas, sistemas, plataformas, aplicaciones, servicios y velocidad) provocando cambios de lo uno sobre lo otro. Surge una nueva Red caracterizada como la web de las personas frente a la web de los datos, correspondiente a la versión uno, la Web 1.0 [FUR07].

Se habla por tanto de una supuesta blogocultura<sup>13</sup>, que se sustenta en la creación colectiva y los contenidos abiertos y que parece configurarse como parte de una realidad sociocultural que apunta hacia la Sociedad de la Conversación, definido como “la primera revolución social del siglo XXI”.

Los siguientes principios son los que ahora se deben de tener en cuenta para incursionar dentro del concepto de Web 2.0.

### 2.2.1 SIETE PRINCIPIOS CONSTITUTIVOS DE LAS APLICACIONES WEB 2.0

Los siete principios son [CRP07]:

1. LA WORLD WIDE WEB COMO PLATAFORMA.

Las nuevas compañías ofrecen software gratuito, utilizando a la Web como plataforma. Así, las herramientas y sus contenidos existen en la propia Web y no en la computadora del usuario. El concepto de webtop opuesto al de desktop es apropiado para explicar este fenómeno comercial.

2. APROVECHAR LA INTELIGENCIA COLECTIVA.

En el entorno Web 2.0 los usuarios actúan de la manera que deseen: en forma tradicional y pasiva, navegando a través de los contenidos; o en forma activa, creando y aportando sus contenidos.

---

<sup>13</sup> Personas que tienen un blog o página personal en donde comparten sus ideas y experiencias.

3. LA GESTIÓN DE LA BASE DE DATOS COMO COMPETENCIA BÁSICA.

Este principio tiene una palabra clave: infoware: software más datos. Lo valioso de las aplicaciones Web 2.0 son los datos, ya que en muchos casos el software es un recurso abierto o de fácil implementación.

4. EL FIN DEL CICLO DE LAS ACTUALIZACIONES DE VERSIONES DEL SOFTWARE.

Como se mencionó antes, se rompe el modelo inicial del software cerrado con derechos de uso y bajo el principio de la obsolescencia planificada, para pasar al uso del software como servicio gratuito, corriendo en la propia Web, y en combinación con los datos.

5. MODELOS DE PROGRAMACIÓN LIGERA. BÚSQUEDA DE LA SIMPLICIDAD.

Los modelos de programación ligera ponen énfasis en la reducción de la complejidad, donde menos es más, evitando las excesivas especificaciones y funciones del software empaquetado. ¿Qué porcentaje de funciones del Microsoft Word se utilizan?

6. EL SOFTWARE NO LIMITADO A UN SOLO DISPOSITIVO.

La utilización de los productos de la Web 2.0 no se limita a las computadoras. Los teléfonos móviles de tercera generación (3G) empezaron a ocupar espacios hasta ahora sólo reservados a las PC's.

7. EXPERIENCIAS ENRIQUECEDORAS DEL USUARIO.

Se deja de lado a las clásicas herramientas para producir webs, como Dreamweaver y Flash, ahora se lo realizan en Ajax, Ruby on Rails o BitTorrent para los productos Web 2.0, además se utilizan blogs, por su fácil uso e instalación.

## 2.3 MODELO VISTA CONTROLADOR

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el controlador es el Sistema de Gestión de Base de Datos y el modelo es el modelo de datos [DHH06].

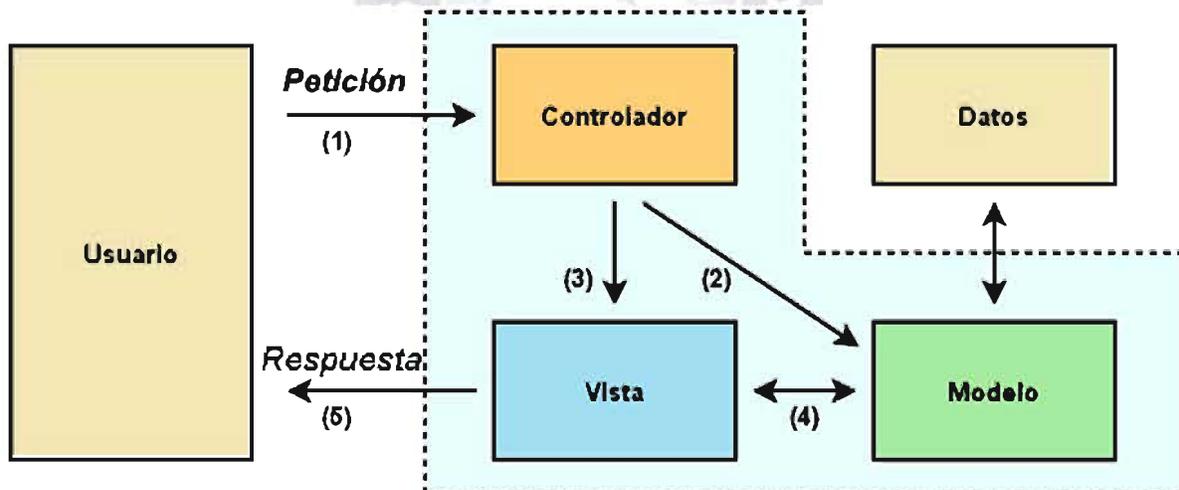
**MODELO:** Esta es la representación específica de la información con la cual el sistema opera además define la lógica de negocio de la aplicación, por ejemplo: Calculando si hoy es el cumpleaños del usuario o los totales, impuestos o portes en un carrito de la compra.

**VISTA:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

**CONTROLADOR:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

### 2.3.1 FLUJO DE ACCIONES DENTRO DEL MVC

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente (Figura 2.5):



**FIGURA 2.5** ARQUITECTURA DEL PATRON MODELO-VISTA-CONTROLADOR

FUENTE: WIKIPEDIA.COM

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.

3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

## 2.4 REPRESENTATIONAL STATE TRANSFER (REST)

La Transferencia de Estado Representacional (Representational State Transfer) o REST es una técnica de diseño de software para sistemas hipermedia como la World Wide Web<sup>14</sup>. El término se originó en el año 2000, en una tesis doctoral<sup>15</sup> sobre la web escrita por Roy

---

<sup>14</sup> Denominación de la telaraña de internet.

<sup>15</sup> "Architectural Styles and the Design of Network-based Software Architectures", University of California, Irvine.

Fielding, uno de los principales autores de la especificación del protocolo HTTP<sup>16</sup> y ha pasado a ser recientemente utilizado por la comunidad de desarrollo web [WIK07].

Es un hecho olvidado por muchos desarrolladores web de hoy que el protocolo HTTP puede hacer algo más que GETs y POSTs. Sin embargo, esto no resulta tan sorprendente si consideramos que los navegadores tan solo soportan esas dos peticiones. GET y POST son los tipos de peticiones HTTP que se suelen transmitir del cliente a servidor, pero el protocolo HTTP también define los métodos PUT y DELETE que, en teoría, se deberían usar para crear o borrar un recurso en la web [WRB07].

Un ejemplo claro es el de utilizar la Urls como en la tabla 2.2:

Acción	Tradicional		REST	
	Método http	URL	Método http	URL
Crear	POST	www.sitio.com/persona/create	POST	www.sitio.com/persona
Mostrar	GET	www.sitio.com/persona/show/1	GET	www.sitio.com/persona/1
Actualizar	POST	www.sitio.com/persona/update/1	UPDATE	www.sitio.com/persona/1
Destruir	POST	www.sitio.com/persona/destroy/1	DELETE	www.sitio.com/persona/1

**TABLA 2.2** COMPARACION ENTRE EL DISEÑO TRADICIONAL Y REST

REST ayuda en el diseño del sitio, lo cual permite ser más simple y usable para el usuario.

## 2.5 AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas se ejecutan en el cliente, es decir, en el navegador de los usuarios, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

Si bien el nombre del mismo representa el manejo y envío asíncrono de archivos XML, también utiliza HTML, que es un nivel inferior del concepto de XML, también otros tipos de archivos de descripción de datos y hasta archivos de texto plano.

<sup>16</sup> Hiper text transfer protocol, Protocolo de transferencia de Hiper Texto, protocolo utilizado para el internet.

AJAX es una combinación de tres tecnologías ya existentes:

- ✓ **XHTML** (o HTML) y hojas de estilos en cascada (**CSS**) para el diseño que acompaña a la información.
- ✓ Document Object Model (**DOM**<sup>17</sup>) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- ✓ El objeto **XMLHttpRequest** para intercambiar datos asincrónicamente con el servidor web.

Dentro del siguiente gráfico se muestra el proceso normal contra el modelo ajax, veamos que se añade una subcapa dentro del cliente que permite las llamadas asíncronas, Figura 2.6.

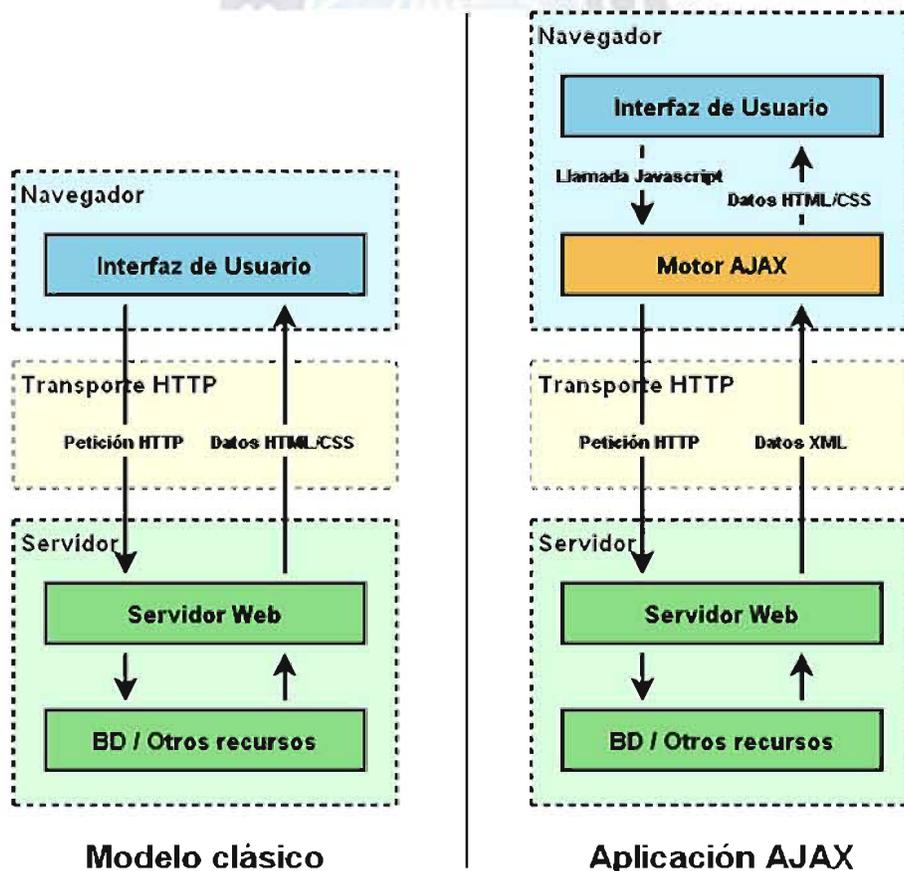


FIGURA 2.6 MODELO TRADIONAL Y AJAX

<sup>17</sup> Representación en árbol jerárquico de las etiquetas Html, que puede ser accedido mediante código javascript.

## 2.5.1 NAVEGADORES WEB QUE PERMITEN AJAX

- Navegadores basados en Gecko como Mozilla, Mozilla Firefox y Netscape versión 7.1 y superiores
- Microsoft Internet Explorer para Windows versión 5.0 y superiores.
- Navegadores Konqueror versión 3.2 y superiores, Apple Safari versión 1.2 y superiores.
- Opera versión 8.0 y superiores, incluyendo Opera Mobile Browser versión 8.0 y superiores

## 2.6 RUBY ON RAILS

Ruby on Rails, también conocido como RoR o Rails es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros frameworks y con un mínimo de configuración. El lenguaje de programación Ruby permite la metaprogramación<sup>18</sup>, de la cual Rails hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible. Rails se distribuye a través de RubyGems<sup>19</sup>, que es el formato oficial de paquete y canal de distribución de librerías y aplicaciones Ruby [WIK07].

### 2.6.3 ARQUITECTURA MVC DE RAILS

Las piezas de la arquitectura Modelo Vista Controlador en Ruby on Rails son las siguientes:

---

<sup>18</sup> Permitir mediante código el poder crear otro programa o algunas funciones, software que crea software.

<sup>19</sup> Paquetes adicionales que maneja el lenguaje de programación Ruby.

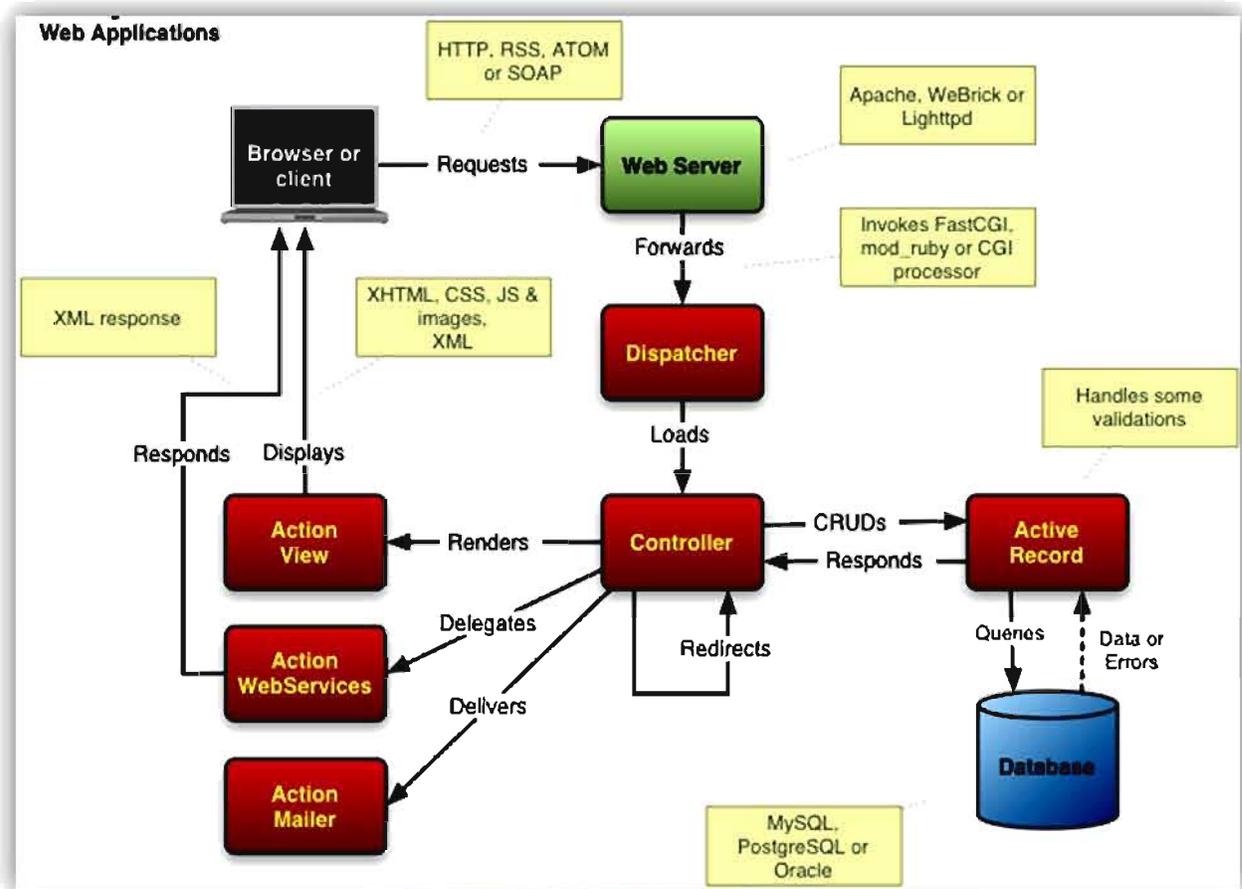


FIGURA 2.7 MODELO DE EJECUCION DE LA PLATAFORMA RUBY ON RAILS

FUENTE: RUBYONRAILS.ORG

## 2.6.4 MÓDULOS DEL FRAMEWORK RUBY ON RAILS

Ruby on Rails por ser una plataforma para desarrollo web incorpora el patrón de arquitectura MVC, además Rails ofrece otros módulos, como Action Mailer (para enviar correo electrónico) y Action Web Service para dar soporte a SOAP y XML-RPC.

Rails es anfitrión del framework Prototype en Javascript (una serie de herramientas que proporcionan llamadas Ajax y otra funcionalidad habitual en las tareas cliente-servidor) lo que permite el desarrollo de aplicaciones web acopladas a ajax .

## 2.6.5 SOPORTE DE SERVIDORES WEB

Para desarrollo y pruebas, se utiliza el servidor web ligero WEBrick, incluido con Ruby. Para su uso en producción se usan Apache o Lighttpd con FastCGI. Sobre Apache, mod\_ruby puede mejorar considerablemente el rendimiento, aunque su uso no se recomienda porque no es seguro utilizar múltiples aplicaciones RoR sobre Apache.

## 2.6.6 SOPORTE DE BASES DE DATOS

Dado que la arquitectura Rails favorece el uso de bases de datos se recomienda usar un SGBD para almacenamiento de datos. Rails soporta la librería SQLite si no es posible emplear una base de datos. El acceso a la base de datos es totalmente abstracto desde el punto de vista del programador, y Rails gestiona los accesos a la base de datos automáticamente (aunque, si se necesita, se pueden hacer consultas directas en SQL) Rails intenta mantener la neutralidad con respecto a la base de datos, la portabilidad de la aplicación a diferentes sistemas de base de datos y la reutilización de bases de datos preexistentes. Sin embargo, debido a la diferente naturaleza y prestaciones de los SGBDs el framework no puede garantizar la compatibilidad completa. Se soportan diferentes SGBDs, incluyendo MySQL, PostgreSQL, SQLite, IBM DB2, Oracle y Microsoft SQL Server.

## 2.7 SUBVERSION

Subversion es un sistema de control de versiones libre y de código fuente abierto. Es decir, Subversion maneja ficheros y directorios a través del tiempo, asignando para esto un número único ID llamado revisión. Hay un árbol de ficheros en un repositorio central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Esto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios entre revisiones, mediante su identificador o fecha de revisión. En este aspecto, mucha gente piensa en los sistemas de versiones como en una especie de “máquina del tiempo” [SUB07].

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. A cierto nivel, la capacidad para que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones. Y puesto que el trabajo se encuentra bajo el control de versiones, no hay razón para temer por que la calidad del mismo vaya a verse afectada por la pérdida de ese conducto único—si se ha hecho un cambio incorrecto a los datos, simplemente deshaga ese cambio.

Algunos sistemas de control de versiones son también sistemas de administración de configuración de software. Estos sistemas son diseñados específicamente para la administración de árboles de código fuente, y tienen muchas características que son específicas del desarrollo de software—tales como el entendimiento nativo de lenguajes de programación, o el suministro de herramientas para la construcción de software. Sin embargo, Subversion no es uno de estos sistemas. Subversion es un sistema general que puede ser usado para administrar cualquier conjunto de ficheros. Para usted, esos ficheros pueden ser código fuente—para otros, cualquier cosa desde la lista de la compra de comestibles hasta combinaciones de vídeo digital y más allá.

## 2.7 MODELO DE TRABAJO EN SUBVERSION COPIAR-MODIFICAR-MEZCLAR

Subversion y otros sistemas de control de versiones utilizan un modelo del tipo **copiar-modificar-mezclar** como alternativa a bloquear un archivo modificado. En este modelo, el cliente de cada usuario se conecta al repositorio del proyecto y crea una copia de trabajo personal—una réplica local de los archivos y directorios del repositorio. Los usuarios pueden entonces trabajar en paralelo, modificando sus copias privadas. Finalmente, todas las copias privadas se combinan (o mezclan) en una nueva versión final. El sistema de control de versiones a menudo ayuda con la mezcla, pero en última instancia es un ser humano el responsable de hacer que esto suceda correctamente.

## 2.8 MODELO DE PROCESO DE LA METODOLOGÍA RUP INCORPORANDO TÉCNICAS DE DESARROLLO DE SOFTWARE PROFESIONAL Y GESTIÓN DEL PROYECTO SCRUM.

### 2.8.1 INTRODUCCIÓN

Como una mejora a la metodología RUP se abarcaran normalmente los distintos artefactos que se necesiten para la captura de requisitos, el análisis y el diseño, pero para la implementación y el despliegue se recurrirán a técnicas y uso de tecnologías que ayudaran al desarrollo del proyecto de manera más profesional, elevando la calidad del software y por igual la calidad del desarrollador mismo, todo lo anterior ya implementado dentro de grandes empresas de desarrollo del software.

Asimismo se debe de hacer notar que se combinara la tecnología con la usabilidad dando como resultado un producto mucho más aceptable para el usuario final. Por las características del equipo que desarrollara el proyecto, se han incluido muy pocos artefactos, roles y actividades de la metodología, manteniendo los más esenciales y descriptivos que facilitaran el análisis y diseño del producto.

### 2.8.2 FASE DE INICIO

#### 2.8.2.1 REQUISITOS

El modelo de requisitos tiene como objetivo delimitar el sistema y capturar la funcionalidad que debe ofrecer desde la perspectiva del usuario. Este modelo puede funcionar como un contrato entre el desarrollador y el cliente o usuario del sistema, y por lo tanto proyecta lo que el cliente desea según la percepción del desarrollador. Por lo tanto, es esencial que los clientes comprender este modelo [JRB99].

El propósito del modelo de requisitos es comprender completamente el problema y sus implicaciones. Todos los modelos no solamente se verifican contra el modelo de requisitos, sino que también se desarrollan directamente en él. El modelo de requisitos sirve también como base para el desarrollo de las instrucciones operacionales y los manuales ya que todo lo que el sistema debe de hacer se describe aquí desde la perspectiva del usuario. El modelo de requisitos no es un proceso mecánico, el analista debe interactuar constantemente con el cliente para completar la información faltante y así clarificar ambigüedades e inconsistencias.

#### **a) Descripción del problema**

La primera actividad que es aconsejable cumplir es llevar una pequeña descripción del problema, esta es bastante preliminar de las necesidades que sirve solamente como un punto de inicio para comprender los requisitos del sistema.

#### **b) Enumerar los requisitos**

Durante la vida del sistema, los clientes, usuarios, analistas y desarrolladores aparecen con muchas buenas ideas que podrían convertirse en verdaderos requisitos [JBR99].

Cada una estas características lleva un nombre corto y una breve explicación o definición y un conjunto de valores tales como estado, coste, prioridad, nivel de riesgo.

### 2.8.2.2 CASOS DE USO

Un caso de uso describe —desde el punto de vista de los actores—, un grupo de actividades de un sistema que produce un resultado concreto y tangible.

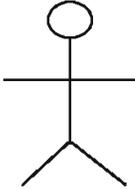
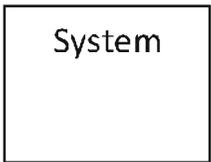
Los casos de uso son descriptores de las interacciones típicas entre los usuarios de un sistema y ese mismo sistema. Representan el interfaz externo del sistema y especifican qué requisitos de funcionamiento debe tener este (recuerde, únicamente el qué, nunca el cómo).

Cuando se trabaja con casos de uso, es importante tener presentes algunas sencillas reglas:

- Cada caso de uso está relacionado como mínimo con un actor.

- Cada caso de uso es un iniciador (es decir, un actor)
- Cada caso de uso lleva a un resultado relevante (un resultado con “valor intrínseco”)

### 2.8.2.2.1 NOTACIÓN

	<b>Actor</b>	_____	<b>Asociación de comunicación</b>
	<b>Caso de uso</b>	----->	<b>Generalización</b>
	<b>Límite del sistema</b>	<<extends>> ----->	<b>Extensión</b> Realizar una interacción adicional. El caso de uso que extiende describe un comportamiento opcional del sistema
		<<include>> ----->	<b>Inclusión</b> Un caso de uso base incorpora explícitamente a otro caso de uso en un lugar especificado en dicho caso base.

**TABLA 2.3 NOTACION DE CASOS DE USO**

Parte fundamental del modelo de casos de uso es una descripción de textual detallada de cada uno de los actores y las descripciones de casos de uso con reseñas textuales del caso de uso. Para documentar los actores encontrados, como los casos de uso se usará el siguiente formato:

<b>Actor:</b>	Nombre del actor.
<b>Casos de uso:</b>	Nombre de los casos de uso en los que participa.
<b>Tipo:</b>	Primario o Secundario.
<b>Descripción:</b>	Breve descripción del actor.

**TABLA 2.4 DESCRIPCION DE ACTORES**

**FUENTE:** INGENIERIA DE SOFTWARE ORIENTADA A OBJETOS, 2002

<b>CU-01: NOMBRE DEL CASO DE USO</b>	
<b>Frecuencia de Uso:</b>	Siempre, A veces o
<b>Actores:</b>	Actores primarios o secundarios que interactúan en los casos de uso
<b>Propósito:</b>	Razón de ser del caso de uso.
<b>Descripción:</b>	Breve descripción del caso de uso.

<b>Prerequisitos:</b>	Condiciones que deben satisfacerse para poder ejecutar el caso de uso
<b>Escenario Principal de Éxito:</b>	El flujo de eventos más importante del caso de uso, donde dependiendo de las acciones de los actores.
<b>Escenario de Extensiones Alternativas:</b>	Flujo alternativo de subrutinas y excepciones.
<b>Post Condiciones:</b>	Condición cumplida luego de ejecutar el caso de uso.

**TABLA 2.5** DESCRIPCION DE CASOS DE USO

**FUENTE:** INGENIERIA DE SOFTWARE ORIENTADA A OBJETOS, 2002

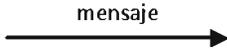
### 2.8.3 FASE DE ELABORACIÓN

Dentro de esta fase se realiza el análisis y el diseño de la solución a partir de los requerimientos, sin llegar a la implementación.

#### 2.8.3.1 ANÁLISIS

##### 2.8.3.1.1 DIAGRAMA DE SECUENCIA

El diagrama de secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un diagrama de secuencia se modela para cada caso de uso. Mientras que el diagrama de caso de uso permite el modelado de una vista 'business' del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes pasados entre los objetos.

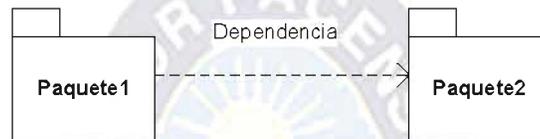
Un objeto se muestra como una caja en parte superior de una línea vertical punteada	
Las activaciones aparecen cuando esta activo un método, ya sea porque está efectuando una operación o porque está esperando la respuesta.	
Cada mensaje se representa mediante una flecha entre las líneas de vida de dos objetos	

**TABLA 2.6** NOTACION DE LOS DIAGRAMAS DE INTERACCION

**FUENTE:** INGENIERIA DE SOFTWARE ORIENTADA A OBJETOS, 2002

### 2.8.3.1.2 DIAGRAMA DE PAQUETES

Rup introduce la noción de un paquete como el ítem universal para agrupar elementos, permitiendo a los modeladores subdividir y categorizar sistemas. Los paquetes pueden ser usados en cualquier nivel, desde el nivel más alto, donde son usados para subdividir el sistema en dominios, hasta el nivel más bajo, donde son usados para agrupar casos de uso individuales, clases o componentes.



**FIGURA 2.8** REPRESENTACION GRAFICA DE PAQUETES

## 2.8.4 FASE DE IMPLEMENTACIÓN

### 2.8.4.1 DIAGRAMA DE CLASES

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas “estáticos” porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases “conocen” a qué otras clases o qué clases “son parte” de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

#### A) CLASE

Una clase define los atributos y los métodos de una serie de objetos. Todos los objetos de esta clase (instancias de esa clase) tienen el mismo comportamiento y el mismo conjunto de atributos (cada objeto tiene el suyo propio). En ocasiones se utiliza el término “tipo” en lugar de clase, pero recuerde que no son lo mismo, y que el término tipo tiene un significado más general.

En las clases están representadas por rectángulos, con el nombre de la clase, y también pueden mostrar atributos y operaciones de la clase en otros dos “compartimentos” dentro del rectángulo.

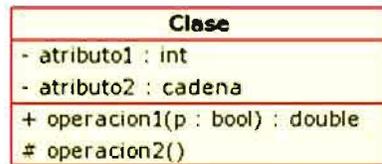


FIGURA 2.9 REPRESENTACIÓN DE UNA CLASE

## B) ATRIBUTOS

En UML, los atributos se muestran al menos con su nombre, y también pueden mostrar su tipo, valor inicial y otras propiedades. Los atributos también pueden ser mostrados visualmente:

- + Indica atributos públicos
- # Indica atributos protegidos
- - Indica atributos privados

## C) OPERACIONES

Las operaciones (métodos) también se muestran al menos con su nombre, y pueden mostrar sus parámetros y valores de retorno. Las operaciones, al igual que los atributos, se pueden mostrar visualmente:

- + Indica atributos públicos
- # Indica atributos protegidos
- - Indica atributos privados

## D) ASOCIACIONES DE CLASES

Las clases se puede tener relaciones (estar asociadas) con otras de diferentes maneras:

## E) GENERALIZACIÓN

La herencia es uno de los conceptos fundamentales de la programación orientada a objetos, en la que una clase “recoge” todos los atributos y operaciones de la clase de la que es heredera, y puede alterar/modificar algunos de ellos, así como añadir más atributos y operaciones propias.

En UML, una asociación de **generalización** entre dos clases, coloca a estas en una jerarquía que representa el concepto de herencia de una clase derivada de la clase base. En UML, las generalizaciones se representan por medio de una línea que conecta las dos clases, con una flecha en el lado de la clase base.



FIGURA 2.10 GENERALIZACIÓN EN UML

## F) ASOCIACIONES

Una asociación representa una relación entre clases, y aporta la semántica común y la estructura de muchos tipos de “conexiones” entre objetos.

Las asociaciones son los mecanismos que permite a los objetos comunicarse entre sí. Describe la conexión entre diferentes clases (la conexión entre los objetos reales se denomina conexión de objetos o **enlace**).

Las asociaciones pueden tener un papel que especifica el propósito de la asociación y pueden ser unidireccionales o bidireccionales (indicando si los dos objetos participantes en la relación pueden intercambiar mensajes entre sí, o es únicamente uno de ellos el que recibe información del otro). Cada extremo de la asociación también tiene un valor de multiplicidad, que indica cuántos objetos de ese lado de la asociación están relacionados con un objeto del extremo contrario.

En UML, las asociaciones se representan por medio de líneas que conectan las clases participantes en la relación, y también pueden mostrar el papel y la multiplicidad de cada

uno de los participantes. La multiplicidad se muestra como un rango [mín...máx] de valores no negativos, con un asterisco (\*) representando el infinito en el lado máximo.



**FIGURA 2.11** ASOCIACION DE CLASES

#### 2.8.4.2 PROTOTIPADO DE NAVEGACIÓN

El objetivo de un prototipo de papel no es probar o verificar lo bonito que es el diseño, sino que se trata de verificar si los usuarios son capaces de realizar sus tareas con la interfaz propuesta. La utilización de esta técnica de prototipado no precisa incorporar avances tecnológicos, solo es necesario que capture la funcionalidad del sistema y que comunique la información y sus interacciones adecuadamente.

Esta técnica consiste en dibujar en un papel, sin entrar en detalles estéticos, las interfaces que se van a probar y valorar. Los diferentes estados de la interfaz se van dibujando en hojas separadas y mediante un proceso de ordenación que el diseñador determina permite que el usuario final interactúe con este material simulando el funcionamiento del sistema

#### 2.8.4.3 DISEÑO DEL MODELO DE RECURSOS WEB MEDIANTE REST

Mediante el diseño del modelo de recursos podemos asignar el tipo de recursos que consumirá cada cliente dentro de la aplicación, mediante la tabla.

Url	Recurso	Acciones	Soporte
<b>Rol de usuario</b>			
URL	Tipo de recurso asignado	Acciones permitidas	Tipos de recursos que soporta la URL

**TABLA 2.7** DESCRIPCION DE ASIGNACION DE RECURSOS REST DENTRO DE UNA APLICACION WEB

#### 2.8.5 GESTIÓN DEL PROYECTO

Como una manera de complementar la metodología RUP, por que la metodología Rup nos responde a preguntas del tipo ¿qué? y no del tipo ¿cómo?, se maneja una novedosa metodología de gestión de proyectos, utilizada en grandes compañías de desarrollo de software<sup>20</sup> llamada Scrum, ver Anexo B.

El desarrollo del sistema se subdividirá en sprints<sup>21</sup> cortos (2 a 4 semanas), al finalizar cada sprint se debe de presentar las nuevas funcionalidades implementadas del software, esto permite dar seguridad al Dueño o Product Owner y mas comunicación dentro del equipo de desarrollo.

#### 2.8.5.1 DEFINICIÓN DE ROLES

La definición de roles se la realiza mediante el listado realizado en el anexo A.

#### 2.8.5.2 DESARROLLO DE LOS SPRINTS

Dentro de la metodología SCRUM, la fase de desarrollo es gradual e iterativo al termino de cada iteración llamada Sprint, se presenta una nueva funcionalidad real dentro del sistema, esto conlleva a tener una estrecha relación con los interesados del sistema, un sprint tiene una duración de dos a cuatro semanas, además tiene una serie de fases las cuales ahora detallaremos:

#### 2.8.5.3 SPRINT BACKLOG

El Sprint Backlog corresponde con un listado de nuevas funcionalidades que se implementaran de acuerdo a los requerimientos o product Backlog. Es decir, del Product Backlog se saca una o más tareas que van a formar parte del Sprint Backlog.

Esta es la plantilla que se genera en cada Sprint Backlog:

---

<sup>20</sup> Microsoft, Google

<sup>21</sup> Carreras a gran velocidad.

PILA DEL SPRINT				
Backlog ID	Tarea	Tipo	Estado	Responsable
	Descripción de la tarea 1	Análisis	En curso	Luis
	Descripción de la tarea 2	Prototipado	Pendiente	Luis
	Descripción de la tarea 3	Pruebas	Terminada	Luis
	Descripción de la tarea 4	Codificación	Terminada	Elena

**Tabla 2.8** Modelo de repartición de tareas dentro de un sprint

Fuente: Naveganapolis.com

#### 2.8.5.4 MODELO DE PRUEBAS

Para cada clase encontrada dentro del proyecto se debe de crear las llamadas pruebas unitarias y de persistencia, los cuales realmente dan a conocer que realmente las clases elaboradas para el sistema están funcionando a la perfección.

2.8.5.5 PRUEBAS UNITARIAS.- Estas pruebas utilizan los métodos generados para cada clase, evaluando cada método y luego comparando con un resultado ya calculado, si existe un error envía un mensaje de resultado incorrecto.

#### 2.8.6 CALIDAD DEL SOFTWARE

##### 2.8.6.1 FACTORES DE CALIDAD

Para determinar la calidad de la aplicación basadas en WEB se debe de considerar los diferentes enfoques y establecer que el punto de vista del usuario puede ser más importante que cualquier discusión técnica de la calidad del aplicativo Web.

Todas las características generales de calidad del software se aplican a la aplicación web. Sin embargo las más relevantes de dichas características con la facilidad por parte del usuario y del cliente es de: Uso, Funcionalidad, confiabilidad, eficiencia y facilidad de mantenimiento, proporcionando una base útil para valorar la calidad de los sistemas basados en la Web.

##### 2.8.6.2 MÉTRICA DE CALIDAD DEL MODELO DE MC-CALL

La iso-9126 plantea un modelo normalizado que permite evaluar y comparar productos sobre la base:

La calidad queda definida a un alto nivel de abstracción por seis características:

- ✓ **Funcionalidad:** Las funciones satisfacen las necesidades declaradas o implícitas.
- ✓ **Fiabilidad:** Capacidad de un sistema para mantener su nivel de rendimiento.
- ✓ **Usabilidad:** Esfuerzo necesario para el uso y la valoración individual de tal uso por parte de un conjunto de usuarios.
- ✓ **Portabilidad:** Es la capacidad de un sistema para ser transferido de un sistema a otro
- ✓ **Mantenibilidad:** Es el esfuerzo necesario para realizar modificaciones específicas
- ✓ **Eficiencia:** Es la relación entre el nivel de prestaciones de un sistema y el volumen de recursos utilizados en condiciones declaradas [ISO 9126: 1991]

Cada factor de calidad (FC) se puede obtener como combinación de una o varias métricas:

$$FC = C_1 M_1 + \dots + C_N M_N \quad \text{ECUACIÓN 2.1}$$

Donde:

**FC:** Es el factor de calidad.

**Cn:** Coeficiente de regresión.

**Mn:** Métricas que afectaran al factor de calidad.

Estas métricas se medirán de forma subjetiva con el siguiente esquema de ponderación de 0 a 10 (0 = Bajo, 5 = Medio y 10 = Alto).

Factor de calidad	Criterio	Ponderación
<b>Funcionalidad</b>	<b>Idoneidad</b>	<b>7</b>
	<b>Corrección</b>	<b>8</b>
	<b>Interoperabilidad</b>	<b>7</b>
	<b>Conformidad</b>	<b>9</b>
	<b>Seguridad</b>	<b>8</b>
<b>Confiabilidad</b>	<b>Madurez</b>	<b>8</b>
	<b>Tolerancia a Fallas</b>	<b>8</b>
	<b>Facilidad de recuperación</b>	<b>9</b>
<b>Usabilidad</b>	<b>Precisión</b>	<b>8</b>
	<b>Consistencia</b>	<b>9</b>
	<b>Tolerancia a fallas</b>	<b>9</b>
	<b>Modularidad</b>	<b>10</b>
	<b>Simplicidad</b>	<b>10</b>
<b>Eficiencia</b>	<b>Tiempo de uso</b>	<b>8</b>
	<b>Recursos utilizados</b>	<b>8</b>
<b>Facilidad de mantenimiento</b>	<b>Facilidad de análisis</b>	<b>9</b>
	<b>Facilidad de cambio</b>	<b>9</b>
	<b>Estabilidad</b>	<b>8</b>

	<b>Facilidad de prueba</b>	<b>9</b>
<b>Portabilidad</b>	<b>Facilidad de instalación</b>	<b>9</b>
	<b>Facilidad de ajuste</b>	<b>9</b>
	<b>Facilidad de adaptación al cambio</b>	<b>10</b>

**TABLA 2.9 FACTORES Y METRICAS**

**FUENTE:** INGENIERÍA DE SOFTWARE, 2002

La calidad del software se calculara en función de los factores considerados y para cada factor las métricas establecidas, promediando los porcentajes de los factores de calidad obtenidos. Siendo el factor de calidad:

$$QT = \sum FACCAL / N^{\circ} \text{ DE FC} \quad \text{ECUACIÓN 2.2}$$

#### 2.8.6.2 MÉTRICAS BASADAS EN LA FUNCIÓN

La métrica de punto función (PF), se usa de manera efectiva como medio para medir la funcionalidad que entrega un sistema. El PF se deriva empleando una relación empírica basada en medidas contables del dominio de la información del software y las evaluaciones de la complejidad de este.

Sea los valores de dominio de la información los siguientes:

- Número de entradas externas (EE)
- Número de salidas externas (SE)
- Número de consultas externas (CE)
- Número de archivos lógicos internos (ALI)
- Numero de archivos interfaz externos (AIE)

A cada conteo se le asocia un valor de complejidad, no obstante la determinación de la complejidad es un poco subjetiva.

Para calcular los puntos de función se usa la siguiente ecuación:

$$PF = \text{CONTEO TOTAL} * [0.65 + 0.01 * \sum(F_i)] \quad \text{ECUACIÓN 2.3}$$

Donde el Conteo total es la suma de de todas las entradas de PF y  $F_i$  son factores de ajuste de valor. Esta métrica poseen un modelo de modelo de valoración entre cero (0) y cinco (5), y

por decisión del equipo de trabajo, se puede asumir una valoración en porcentajes como se muestra en la tabla 2.9 siguiente:

	No influencia	Ninguna	0%	0% - 10%
1	Incidental	Insignificante	1% – 20%	11% – 20%
2	Moderado	Moderada	21% – 40%	21% – 30%
3	Medio	Media	41% – 60%	31% – 40%
4	Significativo	Significativa	61% – 80%	41% – 50%
5	Esencial	Fuerte	81% – 100%	> 50%

**TABLA 2.10 VALORACION METRICA PUNTO FUNCION**

Esta valoración es usada para calificar los siguientes 14 puntos de evaluación:

Nº	Preguntas
1	¿Requiere el sistema copias de seguridad y de recuperación fiables?
2	¿Se requiere de comunicación de datos?
3	¿Existen funciones de procesamiento distribuido?
4	¿Es crítico el rendimiento?
5	¿Se ejecutara el sistema en un entorno operativo existente y frecuentemente utilizado?
6	¿El sistema requiere entradas de datos interactiva?
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples o variadas operaciones?
8	¿Se actualizan los archivos maestros en forma interactiva?
9	¿Son complejas las entradas, salidas, archivos o peticiones?
10	¿Es complejo el procesamiento interno?
11	¿Se ha diseñado el código para ser reutilizado?
12	¿Están incluidas en el diseño la conversión y la instalación?
13	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?
14	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?

**TABLA 2.11 VALORACION METRICA PUNTO FUNCION**

Para poder calcular la calidad de software, tomaremos la siguiente relación:

$$\text{CALIDAD} = \text{ERRORES/PF} \quad \text{ECUACIÓN 2.4}$$

### 2.8.6.3 USABILIDAD CON AJAX

La tecnología Ajax es bastante reciente y se utilizara dentro del proyecto debemos de, no existe una metodología o control de calidad para esta tecnología se optara por describir las siguientes recomendaciones de desarrollo descritas en el anexo B.

## 2.8.7 SEGURIDAD DEL SOFTWARE

La seguridad de las aplicaciones no es un tema que comience en el momento de escribir el código, armar una aplicación segura implica pensar en esta en todo el proceso de desarrollo, desarrollo, podemos comenzar definiendo que es lo que hace la aplicación y como va a ser utilizado.

Hacer consideraciones de seguridad significa de alguna manera ponerse y pensar como el agresor, tratar de encontrar en el diagrama las vulnerabilidades y buscar mecanismos que solucionen estas vulnerabilidades encontradas.

Amenaza	Ejemplos
Inyección de SQL	Incluyendo comandos de SQL al ingresar datos para que el sistema los procese.
Script entre sitios	Usando scripts maliciosos del lado del cliente.
Modificación del ingreso.	Cambiando valores de campos ocultos
Robo de claves.	Usando un investigador de paquetes (sniffer) para obtener claves de acceso o cookies no encriptados.
Reemplazo de sesión.	Uso de cookie de sesión robado, para acceder al sitio.
Reemplazo de identidad.	Uso de la cookie de autenticación por formulario para hacerse pasar por otro.
Revelado de información	Mostrar al cliente el seguimiento de la pila cuando sucede un error

**TABLA 2.12 AMENAZAS COMUNES DENTRO DE APLICACIONES WEB**

Para el modelado de amenazas se procede a la clasificación y puntuación de de amenazas basados en el enfoque del modelo DREAD que facilita la puntuación de peligro asociado al uso del sitio. (Damage potential, Reproducibility, Exploitability, Affect users, Discoverability)

### 2.8.7.1 MÉTODO DE PUNTUACIÓN DREAD

Una vez que tenemos identificada la lista de amenazas, el siguiente paso consiste en puntuarlas de acuerdo al riesgo que suponen. Esto nos permitirá priorizar las actuaciones a efectuar para mitigar el riesgo. Recordemos que, el riesgo se puede cuantificar como el resultado de multiplicar la probabilidad de que la amenaza se produzca, por el daño potencial de esta:

$$\text{RIESGO} = \text{PROBABILIDAD} * \text{DAÑO POTENCIAL. (ECUACIÓN 2.5)}$$

Directiva	Pregunta
Daño potencial.	¿Cuál es el daño que origina si la vulnerabilidad logra ser descubierta?
Facilidad de reproducción	¿Se puede reproducir bajo ciertas circunstancias?
Capacidad de Explotación.	¿Cuán fácil es realizarla?
Usuarios afectados.	¿Cuántos pueden verse afectados?
Dificultad para su descubrimiento.	¿Es fácil de encontrar la vulnerabilidad?

**TABLA 2.13** DESCRIPCION DEL MECANISMO DREAD

El método de medición de DREAD es el de cuantificar los distintos elementos de este método en tres del tipo: Alto, medio y bajo, la siguiente tabla muestra el significado de cada elemento que compone el método DREAD.

Vulnerabilidad	Alto	Medio	Bajo
Daño potencial.	El agresor puede obtener datos muy sensibles dañar servidores, etc.	Puede obtener datos sensibles pero casi nada más.	Puede acceder a datos poco importantes.
Facilidad de reproducción	Siempre es posible.	Sucede si se realiza en un corto tiempo.	Raramente se puede hacer.
Capacidad de Explotación.	Puede hacerlo.	Se debe tener ciertos conocimientos.	Tal vez alguna.
Usuarios afectados.	La mayoría.	Algunos	Pocos, si es que hay alguno.
Dificultad para su descubrimiento.	Fácil de ver	Mas de costoso de ver.	Muy difícil de encontrar

**TABLA 2.14** DESCRIPCION DE LAS PONDERACIONES PARA APLICAR DREAD

# Capítulo III

En este capítulo se desarrollara la solución basándonos en el problema planteado, para obtener el producto software que controle la planeación pedagógica y los pagos dentro del Instituto Exclusivo de Idiomas.

Por las características del equipo que desarrollara el proyecto se ha efectuado un recorte dentro de la metodología rup, incluyendo solamente los artefactos más esenciales y descriptivos que facilitaran el análisis y diseño del software. Además se hace uso de una nueva metodología de gestión de proyectos SCRUM.

## 3.1 FASE DE INICIO

### 3.1.1 ANÁLISIS DE REQUISITOS

Dentro de la fase de captura de los requerimientos se elaboraran los siguientes artefactos:

- Requisitos candidatos.
- El Product Backlog.

#### 3.1.1.1 REQUISITOS CANDIDATOS

Pasamos a describir los requisitos candidatos encontrados (ver 2.9.2.1 a) y b))

Nombre	Definición	Estado	Costo	Prioridad	Riesgo
Actividades pedagógicas	Registrar y cancelar actividades pedagógicas	A	M	I	S
Clientes de traducciones	Registrar clientes de traducciones, para un posterior evaluación.	A	B	S	O
Comunicación	Brindar canales de comunicación entre los estudiantes, profesores y secretaria.	I	M	I	O
Control de acceso y seguridad	Gestionar el control mediante asignación de roles, además el manejo de seguridad de la aplicación	A	A	I	S
Disponibilidad	Disponible en cualquier lugar gracias al internet.	A	M	C	O
Estructura en capas	Organización lógica del sistema (MVC)	P	A	C	C
Estudiantes	Registro de estudiantes	A	B	S	O
Horarios disponibles	Registrar disponibilidad de horarios disponibles para profesores.	I	M	S	O

Log	Registrar operaciones realizadas por los usuarios.	I	B	I	S
Mantenimiento y backups	Conjunto de operaciones y/o cuidados necesarios para un adecuado funcionamiento.	I	B	I	O
Multilingüe	Interfaz que permita cambiar de idioma.	A	A	C	S
Notificar	Hacer conocer un evento al usuario mediante un correo electrónico.	I	M	S	O
Pagos	Registrar pagos de profesores y estudiantes.	A	M	I	S
Perfiles	Registro y manejo de perfiles de estudiantes.	I	M	S	O
Reservación	Estudiantes con crédito pueden reservar actividad académica.	A	M	C	S
Traducciones	Registro de traducciones.	I	M	I	O

**Estado:** Propuesto (P), Aprobado (A), Incluido (I)

**Costo:** Alto (A), Medio (M), Bajo (B)

**Prioridad:** Critico (C), Importante (I), Secundario (S)

**Riesgo:** Critico (C), Significativo (S), Ordinario (O)

TABLA 3.1 REQUISITOS CANDIDATOS

Como una manera de complementar la captura de requisitos se elabora el product backlog de SCRUM, que se detalla en el anexo C.

### 3.2.2 CASOS DE USO

#### 3.2.2.1 DIAGRAMA INICIAL



FIGURA 3.1 DIAGRAMA DE CASOS DE USO PRINCIPAL

### 3.2.2.2 DIAGRAMA DE CASOS DE USO: REGISTRO DE ACTIVIDADES PEDAGÓGICAS

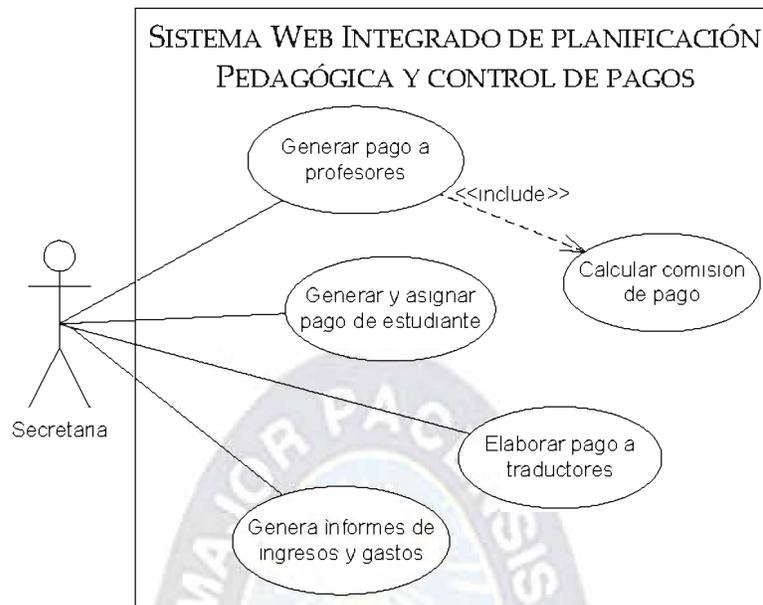
El registro de actividades pedagógicas es cualquier actividad, dentro del instituto, que permita el aprendizaje de idiomas como: español, inglés, aimara y que luego tenga que pagarse al profesor. Por el momento actividad pedagógica se remite a clases individuales o grupales de idiomas, pero también pueden ser reuniones, videos, visitas a museos o lo que vea conveniente el profesor dentro de su programa.



**FIGURA 3.2** DIAGRAMA DE CASOS DE USO REGISTRO DE ACTIVIDADES PEDAGÓGICAS

### 3.2.2.2 DIAGRAMA DE CASOS DE USO GESTIÓN DE PAGOS

Dentro de la gestión de casos de uso abordamos los casos de generar pago a profesores, generar y asignar el pago de estudiante, elaborar pago a traductores y generar informes de ingresos por pagos de estudiantes y por gastos en pagos a profesores.



**FIGURA 3.3** CASOS DE USO REGISTRO Y GENERACION DE PAGOS

### 3.2.2.2 DIAGRAMA DE CASOS DE USO: REGISTRO DE TRADUCCIONES

El siguiente caso de uso muestra las acciones que permitan gestionar traducciones que se realizan dentro del instituto.



**FIGURA 3.4** CASOS DE USO MANEJO DE TRADUCCIONES Y CLIENTES DE TRADUCCIONES

### 3.2.2.1 ACTORES ENCONTRADOS.

Los actores encontrados en los casos de uso de las figuras 3.1 al 3.4, son:

Usuario, Estudiante, Profesor, Secretaria, Traductor y administrador del sistema, los cuales pasamos a describirlos según el formato de la tabla 2.4.

Las siguientes tablas describen los actores encontrados:

<b>Actor:</b>	Estudiante
<b>Casos de uso:</b>	Registra clases, examina horarios de profesores.
<b>Tipo:</b>	Principal
<b>Descripción:</b>	Es la persona que está inscrito y pasando clases dentro del instituto, si tiene crédito suficiente puede registrar clases.

**TABLA 3.2 DESCRIPCION DEL ACTOR ESTUDIANTE**

<b>Actor:</b>	Profesor
<b>Casos de uso:</b>	Registra clases, Llena informe de clases, Registra horarios disponibles.
<b>Tipo:</b>	Principal
<b>Descripción:</b>	Es la persona que dicta clases dentro del instituto, debe de registrar las clases que dicta en el instituto.

**TABLA 3.3 DESCRIPCION DEL ACTOR PROFESOR**

<b>Actor:</b>	Secretaria
<b>Casos de uso:</b>	Examina horarios de profesores, Asigna crédito a estudiante, Realiza pagos a profesores, Genera informes, Registra clases, actualiza crédito de estudiantes.
<b>Tipo:</b>	Principal
<b>Descripción:</b>	Es la persona que está controlando los registros de clases del sistema, y registra los pagos de estudiantes y profesores.

**TABLA 3.4 DESCRIPCION DEL ACTOR SECRETARIA**

<b>Actor:</b>	Traductor
<b>Casos de uso:</b>	Ingreso al sistema, Registra traducciones
<b>Tipo:</b>	Principal
<b>Descripción:</b>	Es la persona que realiza traducciones de documentos en diferentes idiomas dentro del instituto.

**TABLA 3.5 DESCRIPCION DEL ACTOR TRADUCTOR**

### 3.2.2.2 DESCRIPCIÓN DE LOS CASOS DE USO

Las siguientes tablas describen los casos de uso encontrados, siguiendo el formato que se propone en la tabla 2.5:

<b>CU-01: AUTENTICARSE</b>	
<b>Frecuencia de Uso:</b>	Siempre
<b>Actores:</b>	Usuarios
<b>Propósito:</b>	Autenticar al usuario
<b>Descripción:</b>	Identificar, validar al usuario y comprobar realmente quien dice ser mediante los datos ingresados.

<b>Prerequisitos:</b>	Ingresar al sitio.
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa su nombre de usuario y la contraseña.</li> <li>2. Comparar los datos ingresados contra los datos de la base de datos</li> <li>3. Se valida satisfactoriamente al usuario.</li> </ol>
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>• Si no se valida satisfactoriamente entonces se muestra el mensaje de autenticación fallida.</li> </ul>
<b>Post Condiciones:</b>	Usuario autenticado.

**TABLA 3.6 CASO DE USO AUTENTICARSE**

<b>CU-02: REGISTRARSE DENTRO DEL SISTEMA</b>	
<b>Frecuencia de Uso:</b>	A veces
<b>Actores:</b>	Usuarios
<b>Propósito:</b>	Registrar dentro del sistema a un usuario
<b>Descripción:</b>	Registrarse un usuario llenando un formulario con sus datos personales.
<b>Prerequisitos:</b>	Ingresar al portal.
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>1. El usuario elige el vínculo de registrarse al sistema</li> <li>2. Se le presenta al usuario un formulario en blanco para poder llenarlo.</li> <li>3. El usuario debe de llenar datos únicos como ser: nombre de usuario, contraseña y correo.</li> <li>4. Se ejecuta el caso de uso: confirmar registró.</li> </ol>
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>• Si ya existe registrado el usuario o al confirmar la contraseña no es igual o al escribir un correo electrónico ya registrado el sistema le informa que ya existen estos datos y debe de cambiarlos.</li> </ul>
<b>Post Condiciones:</b>	Usuario registrado en el sistema.

**TABLA 3.7 CASO DE USO REGISTRARSE DENTRO DEL SISTEMA**

<b>CU-03: ACTIVAR CUENTA DE USUARIO</b>	
<b>Frecuencia de Uso:</b>	A veces.
<b>Actores:</b>	Usuario.
<b>Propósito:</b>	Confirmar los datos del usuario.
<b>Descripción:</b>	Confirmar mediante el correo electrónico la veracidad de los datos ingresados por el usuario.
<b>Prerequisitos:</b>	Haberse registrado dentro del sistema.
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>1. Una vez realizado el registro del usuario dentro del sistema, se le envía un correo electrónico a la dirección ingresada en el formulario de registro.</li> <li>2. El usuario ingresa en su cuenta de correo electrónico y debe de abrir el correo electrónico enviado por el instituto.</li> <li>3. El usuario debe de ir a la dirección que se autogenera dentro del correo electrónico.</li> <li>4. Una vez dentro del sitio se le informa que ya ha sido registrado dentro del sistema.</li> <li>5. El usuario ahora debe de esperar a que se le asigne crédito mediante secretaria.</li> </ol>
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>• Si el nuevo usuario esta ya inscrito físicamente, la secretaria será el que lo registre y active al usuario.</li> </ul>
<b>Post Condiciones:</b>	Usuario registrado y confirmado dentro del sistema.

**TABLA 3.8 CASO DE USO ACTIVAR CUENTA DE USUARIO**

CU-04: ELABORAR PROGRAMA Y CONTENIDO DEL CURSO	
<b>Frecuencia de Uso:</b>	A veces
<b>Actores:</b>	Profesor
<b>Propósito:</b>	Elaborar y/o modificar el programa de curso de idiomas
<b>Descripción:</b>	Llenar, enviar y/o modificar el programa de estudios que se maneja dentro del instituto a los estudiantes para su seguimiento.
<b>Prerequisitos:</b>	Estar autenticado y tener el rol de profesor.
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>5. El profesor elige la opción cursos.</li> <li>6. Examina los cursos y selecciona el programa y lo baja.</li> <li>7. Modifica el programa y lo vuelve a subir.</li> <li>8. Selecciona a los estudiantes que quiere enviar el programa y selecciona enviar.</li> </ol>
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>• Si ya existe registrado el usuario o al confirmar la contraseña no es igual o al escribir un correo electrónico ya registrado el sistema le informa que ya existen estos datos y debe de cambiarlos.</li> </ul>
<b>Post Condiciones:</b>	Usuario registrado en el sistema.

**TABLA 3.9 ELABORAR PROGRAMA Y CONTENIDO DEL CURSO**

CU-05: REGISTRA ACTIVIDAD PEDAGÓGICA	
<b>Frecuencia de Uso:</b>	Siempre
<b>Actores:</b>	Estudiante, Profesor.
<b>Propósito:</b>	Registrar actividades y/o clases dentro del sistema entre profesores y estudiantes.
<b>Descripción:</b>	Se registra una clase entre un profesor y uno o más estudiantes, evitando choques y notificando vía correo electrónico de la clase registrada.
<b>Prerequisitos:</b>	Se requiere que este autenticado el usuario Que el usuario sea del tipo secretaria, profesor o estudiante con crédito.
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>1. El usuario se le presenta un formulario para el registro de una nueva clase</li> <li>2. Se selecciona la fecha y la hora y el respectivo estudiante y el profesor</li> <li>3. Se graba y envía una notificación al profesor y los estudiantes de la clase.</li> <li>4. Se genera un horario para la clase.</li> </ol>
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>• Si la clase tiene un choque con otra clase se le notifica</li> <li>• Si la clase tiene una duración de menos de una hora no se podrá registrar la clase</li> <li>• Si es el estudiante que quiere registrar la clase debe de tener crédito suficiente para la clase.</li> </ul>
<b>Post Condiciones:</b>	Clase registrada en el sistema.

**TABLA 3.10 CASO DE USO REGISTRA CLASES**

CU-06: ACTUALIZA EL CREDITO DEL ESTUDIANTE	
<b>Frecuencia de Uso:</b>	Siempre
<b>Actores:</b>	Secretaria.
<b>Propósito:</b>	Actualizar el crédito del estudiante
<b>Descripción:</b>	Después de haberse registrado una clase dentro del sistema, el crédito del estudiante se actualiza para tener un seguimiento financiero del estudiante.
<b>Prerequisitos:</b>	Anteriormente se produjo el caso de uso registrar actividades pedagógicas.
<b>Escenario Principal de</b>	<ol style="list-style-type: none"> <li>1. Después de registrada la clase, automáticamente se actualiza el crédito</li> </ol>

<b>Éxito:</b>	del usuario
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>• Ninguno</li> </ul>
<b>Post Condiciones:</b>	El estudiante tiene actualizado su situación financiera

**TABLA 3.11 CASO DE USO ACTUALIZA EL CREDITO DEL ESTUDIANTE**

<b>CU-07: NOTIFICACIÓN DE SUCESOS EN LAS ACTIVIDADES</b>	
<b>Frecuencia de Uso:</b>	Siempre
<b>Actores:</b>	Secretaría.
<b>Propósito:</b>	Notifica cualquier suceso que suceda en una actividad entre los estudiantes, profesores, y administración.
<b>Descripción:</b>	Después de haberse registrado una clase dentro del sistema, se informa mediante un correo electrónico o chat, los distintos sucesos dentro de la actividad, como ser: cambio de horario, cancelación de la actividad, ver informe de la actividad.
<b>Prerequisitos:</b>	Anteriormente se produjo el caso de uso registrar clase
<b>Escenario Principal de Éxito:</b>	1. En cada clase se encuentra la posibilidad de informar sucesos dentro de la actividad.
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>• Ninguno</li> </ul>
<b>Post Condiciones:</b>	Estudiante y profesores saben sobre los sucesos que se realizan dentro de las actividades.

**TABLA 3.12 CASO DE USO ACTUALIZA EL CREDITO DEL ESTUDIANTE**

<b>CU-08: LLENAR Y COMPARTIR EL PERFIL DE USUARIO</b>	
<b>Frecuencia de Uso:</b>	A veces
<b>Actores:</b>	Estudiante
<b>Propósito:</b>	Crear y registrar el perfil del estudiante
<b>Descripción:</b>	Como una manera de evaluar las características del estudiante se le pide que llene un formulario dentro del instituto, luego este está disponible para los profesores del instituto.
<b>Prerequisitos:</b>	Estar autenticado y tener el rol de estudiante.
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>1. El estudiante activa la opción de perfil</li> <li>2. Selecciona llenar hoja de registro</li> <li>3. Enviar el formulario</li> <li>4. Los profesores pueden acceder si se lleno el formulario del estudiante.</li> </ol>
<b>Escenario de Extensiones Alternativas:</b>	
<b>Post Condiciones:</b>	Perfil del estudiante llenado y disponible para los profesores.

**TABLA 3.13 LLENAR Y COMPARTIR EL PERFIL DE USUARIO**

<b>CU-09: ELABORAR Y COMPARTIR EL INFORME DE LA CLASE</b>	
<b>Frecuencia de Uso:</b>	Siempre
<b>Actores:</b>	Profesor
<b>Propósito:</b>	Realizar un informe después de la clase realizada.

<b>Descripción:</b>	Después de pasada la clase, el profesor debe de llenar un pequeño informe sobre la clase, esto para el seguimiento de la clase por parte de profesores y posteriormente realizar los pagos a profesores por actividades realizadas.
<b>Prerequisitos:</b>	Haber registrado una actividad y/o clase dentro del sistema y tener el rol de profesor.
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>1. El profesor se va al sitio y dentro de sus horarios examina los horarios y selecciona la clase deseada.</li> <li>2. Anota los detalles de la clase</li> <li>3. Graba los datos introducidos</li> </ol>
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>• Si la clase todavía no se ha realizado no aparece el informe para llenarlo</li> </ul>
<b>Post Condiciones:</b>	Reporte de la clase realizada

**TABLA 3.14 ELABORAR Y COMPARTIR EL INFORME DE LA CLASE**

<b>CU-10: REGISTRAR DISPONIBILIDAD DE HORARIOS</b>	
<b>Frecuencia de Uso:</b>	Siempre
<b>Actores:</b>	Profesor
<b>Propósito:</b>	Registrar horarios disponibles
<b>Descripción:</b>	Cada profesor debe de registrar sus horarios disponibles, para que estudiantes puedan registrar clases de manera rápida y sencilla.
<b>Prerequisitos:</b>	Estar autenticado dentro del sitio y tener el rol de profesor
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>1. El profesor se dirige a la pestaña para ver sus horarios disponibles.</li> <li>2. Ingresa una nueva fecha y hora del horario.</li> </ol>
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>• Si horario tiene algún choque solamente se reescribe añadiendo el horario anterior.</li> </ul>
<b>Post Condiciones:</b>	Profesor con horarios disponibles.

**TABLA 3.15 CASO DE USO REGISTRA HORARIOS DISPONIBLES**

<b>CU-11: EXAMINA HORARIOS DE PROFESOR</b>	
<b>Frecuencia de Uso:</b>	Siempre
<b>Actores:</b>	Estudiante, Secretaria.
<b>Propósito:</b>	Examinar horarios disponibles para poder pasar clases con profesor.
<b>Descripción:</b>	El usuario examina los horarios disponibles para poder registrar un horario de clases en su horario disponible.
<b>Prerequisitos:</b>	Estar autenticado dentro del sitio y tener el rol de estudiante o secretaria
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la pestaña de profesores, y se le muestra la listado de los profesores del instituto.</li> <li>2. En cada profesor existe una columna de ver horarios disponibles.</li> <li>3. Se muestra el horario disponible, por día, semana.</li> </ol>
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>• Ninguno</li> </ul>
<b>Post Condiciones:</b>	Usuario se informa de los horarios disponibles del profesor.

**TABLA 3.16 CASO DE USO EXAMINA HORARIOS DE PROFESOR**

<b>CU-12: REGISTRAR Y CALCULAR COSTO DE TRADUCCIONES</b>	
<b>Frecuencia de Uso:</b>	Siempre
<b>Actores:</b>	Traductor, Secretaria.
<b>Propósito:</b>	Registrar una traducción realizada dentro del instituto.
<b>Descripción:</b>	Cuando se recepciona una nueva traducción, esta es registrada, para posteriormente calcular el costo de la traducción, evaluando el idioma y el costo por palabra.
<b>Prerequisitos:</b>	Estar autenticado dentro del sistema y tener el rol de secretaria o traductor.
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de registrar nueva traducción.</li> <li>2. Ingresar los datos relevantes de la traducción.</li> <li>3. Tiene la opción de subir el documento a traducir.</li> <li>4. Guarda los datos.</li> </ol>
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>• Ninguno</li> </ul>
<b>Post Condiciones:</b>	Nueva traducción registrada

**TABLA 3.17 CASO DE USO REGISTRAR Y CALCULAR COSTO DE TRADUCCIONES**

<b>CU-13: SEGUIMIENTO Y ENVÍO DE LA TRADUCCIÓN</b>	
<b>Frecuencia de Uso:</b>	Siempre
<b>Actores:</b>	Traductor
<b>Propósito:</b>	Cerrar la traducción, y enviar la traducción realizada al cliente.
<b>Descripción:</b>	Una vez realizada la traducción del documento el traductor realiza el cierre de la traducción, que avala su pago por comisión, a su vez envía el documento por correo electrónico si es necesario.
<b>Prerequisitos:</b>	Tener el rol de traductor y tener asignada o activa la traducción.
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>1. El profesor se va al sitio y dentro de sus horarios examina los horarios y selecciona la clase deseada.</li> <li>2. Anota los detalles de la clase</li> <li>3. Graba los datos introducidos</li> </ol>
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>• Si la clase todavía no se ha realizado no aparece el informe para llenarlo</li> </ul>
<b>Post Condiciones:</b>	Reporte de la clase realizada

**TABLA 3.18 CASO DE USO SEGUIMIENTO Y ENVÍO DE LA TRADUCCIÓN**

<b>CU-14: REGISTRAR Y GESTIONAR CLIENTES</b>	
<b>Frecuencia de Uso:</b>	Siempre
<b>Actores:</b>	Secretaria
<b>Propósito:</b>	Tener una base de datos de clientes para realizar promociones.
<b>Descripción:</b>	Cuando se realiza una traducción se registran los datos del cliente para posteriormente realizar promociones.
<b>Prerequisitos:</b>	Tener el rol de secretaria.
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la pestaña de clientes.</li> <li>2. Registrar los datos del cliente.</li> </ol>

<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>Ninguno.</li> </ul>
<b>Post Condiciones:</b>	Reporte de la clase realizada

**TABLA 3.19 CASO DE USO REGISTRAR Y GESTIONAR CLIENTES**

<b>CU-15: ASIGNA CRÉDITO A ESTUDIANTES</b>	
<b>Frecuencia de Uso:</b>	Siempre
<b>Actores:</b>	Secretaría.
<b>Propósito:</b>	Asignar crédito a estudiantes.
<b>Descripción:</b>	Cada estudiante tiene un seguimiento de su estado financiero, pagos y deudas dentro del instituto, con este caso de uso se asigna crédito al estudiante para que pueda registrar sus clase
<b>Prerequisitos:</b>	Estar autenticado y tener el rol de secretaria.
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>Un estudiante quiere realizar el pago de sus clases,</li> <li>La secretaria entra a la opción nuevo pago</li> <li>Llena el recibo de nuevo pago de estudiante.</li> <li>Imprime el recibo.</li> <li>El usuario está asignado el crédito pagado</li> </ol>
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>Si se ingresa un monto negativo, no puede continuar con la transacción.</li> </ul>
<b>Post Condiciones:</b>	Usuario registrado y confirmado dentro del sistema.

**TABLA 3.20 CASO DE USO ASIGNA CREDITO A ESTUDIANTES**

<b>CU-16: REALIZA PAGOS A PROFESORES</b>	
<b>Frecuencia de Uso:</b>	A veces.
<b>Actores:</b>	Secretaría
<b>Propósito:</b>	Realizar cuando lo requiera pagos por clases pasadas a profesores del instituto.
<b>Descripción:</b>	Un profesor al registrar sus clases pasadas y posteriormente llenar el informe de la clase, se le asigna un porcentaje de ganancia, el cual se suma y se genera el pago respectivo al profesor.
<b>Prerequisitos:</b>	Estar autenticado, tener el rol de secretaria, y el profesor tiene clases pasadas.
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>La secretaria hace el listado de los profesores del instituto.</li> <li>Selecciona al profesor y ver clases pasadas.</li> <li>Selecciona las clases a pagar y ejecuta generar pago.</li> <li>Se imprime la boleta de pago del profesor.</li> </ol>
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>Ninguno</li> </ul>
<b>Post Condiciones:</b>	Usuario registrado y confirmado dentro del sistema.

**TABLA 3.21 CASO DE USO REALIZA PAGOS A PROFESORES**

CU-17: GENERAR INFORMES DE INGRESOS Y GASTOS	
<b>Frecuencia de Uso:</b>	A veces.
<b>Actores:</b>	Secretaria.
<b>Propósito:</b>	Generar informes de ingresos y gastos .
<b>Descripción:</b>	Elaborar informe del total de ingresos por pagos de estudiantes y los gastos por comisiones por pagos de profesores, estos informes por fechas.
<b>Prerequisitos:</b>	Tener el rol de secretaria, y haber registrado pagos de estudiantes y/o comisión de profesores.
<b>Escenario Principal de Éxito:</b>	<ol style="list-style-type: none"> <li>1. El usuario con rol de secretaria, selecciona la opción de reportes.</li> <li>2. Selecciona generar reporte por pagos de profesor.</li> <li>3. Imprime o guarda el reporte.</li> </ol>
<b>Escenario de Extensiones Alternativas:</b>	<ul style="list-style-type: none"> <li>• Ninguno</li> </ul>
<b>Post Condiciones:</b>	Informe de gastos o ingresos.

TABLA 3.22 CASO DE USO GENERAR INFORMES DE INGRESOS Y GASTOS

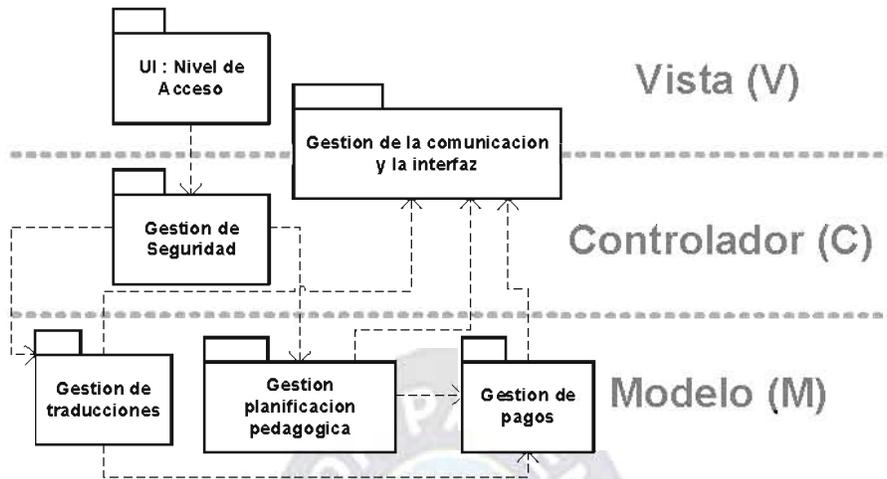
### 3.3 FASE DE ELABORACIÓN

#### 3.3 ANÁLISIS

Durante esta fase se realiza el análisis de los requisitos descritos por los requisitos y los casos de uso especificados en la fase de inicio, el objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que nos ayude a construir el sistema. Para ello haremos uso del diagrama de estados, diagramas de interacción y diagramas de paquetes.

##### 3.3.1 DIAGRAMA DE PAQUETES

Dentro del análisis del sistema es la categorización. Esto es, los diferentes elementos de los modelos de análisis se clasifican de una manera que se empaquetan como una agrupación, añadiendo a esto la arquitectura que se decidió utilizar: Modelo, Vista, Controlador, que separa en tres capas al diseño e implementación del producto.



**FIGURA 3.5** DIAGRAMA DE PAQUETES

### 3.3.2 DIAGRAMA DE SECUENCIAS

Los diagramas de secuencia son modelos que describen la manera en que colaboran grupos de objetos para cierto comportamiento, indican como los eventos causan transiciones de objeto a objeto. A continuación describimos los más importantes.

a. Registro de actividades: este diagrama ()



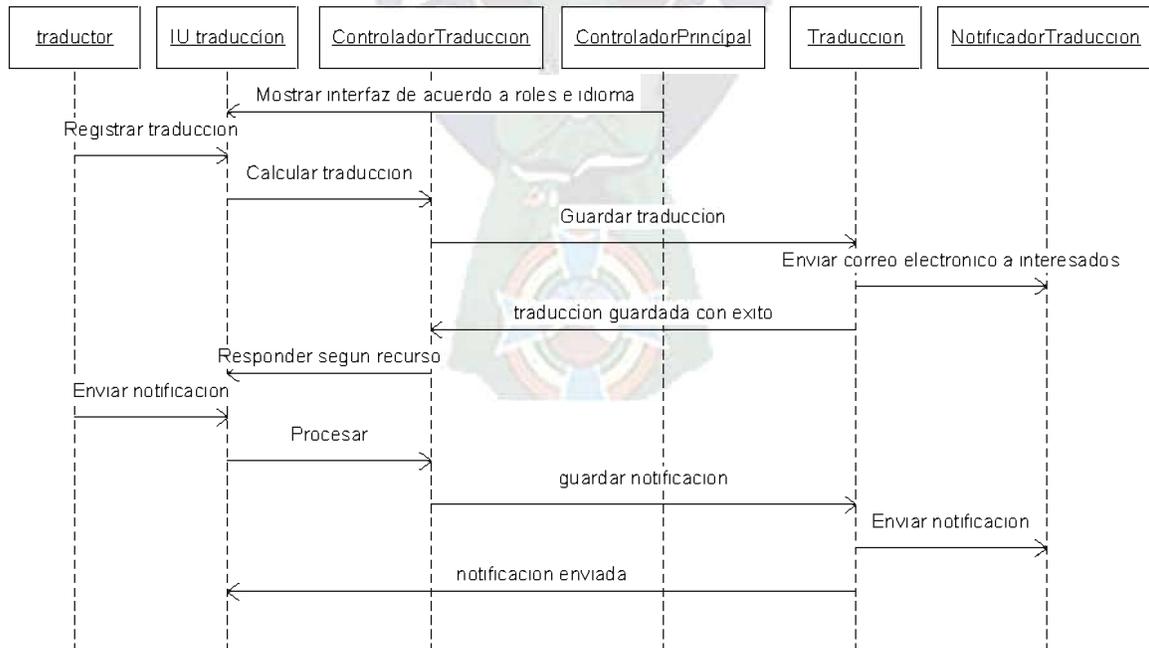
**FIGURA 3.6** DIAGRAMA DE SECUENCIA REGISTRO DE NUEVA ACTIVIDAD PEDAGOGICA

b. Generación de pagos a profesores



**FIGURA 3.7** DIAGRAMA DE SECUENCIA GENERACION DE PAGOS A PROFESORES

c. Registro de traducciones y gestión del desarrollo de traducciones



**FIGURA 3.8** DIAGRAMA DE SECUENCIA REGISTRO DE TRADUCCIONES Y GESTION DE LA TRADUCCION

### 3.4 FASE DE IMPLEMENTACIÓN

#### 3.4.1 DIAGRAMA DE CLASES – SISTEMA WEB INTEGRADO DE PLANIFICACIÓN

##### Y PAGOS

La figura muestra la representación de las clases encontradas y sus relaciones, tomar en cuenta que cada clase cuenta por defecto con su crear, modificar y borrar respectivo, manejando permisos y/o roles, gracias al controlador dentro de la arquitectura MVC:

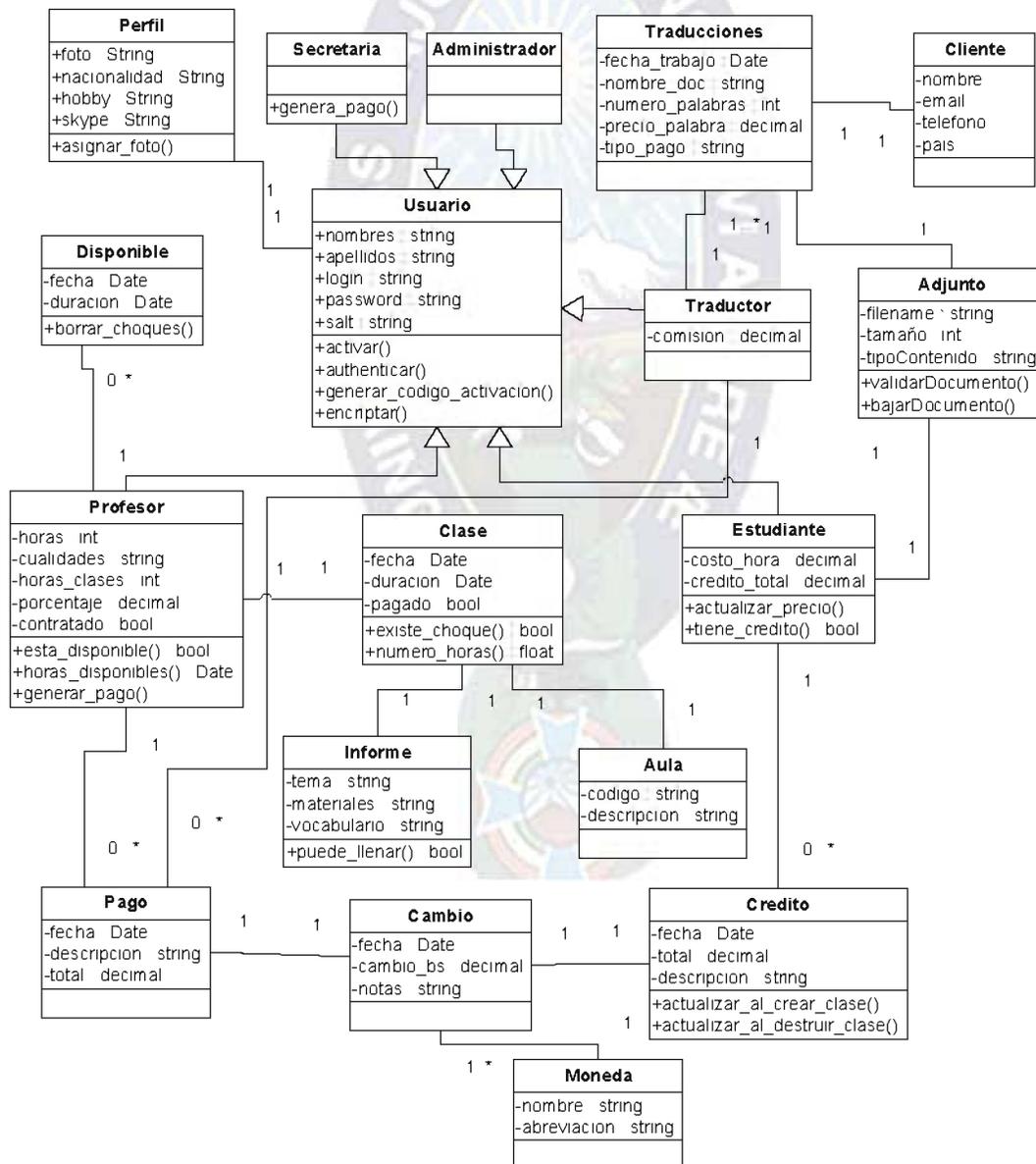


FIGURA 3.9 DIAGRAMA DE CLASES DEL PROYECTO

### 3.4.2 PROTOTIPADO DE NAVEGACIÓN

La tabla representa la distribución del menú principal para secretaria del instituto, presentado al usuario.

MENU	SUBMENU	DESCRIPCION
USUARIOS	NUEVO	Registra un nuevo usuario dentro del sistema, además se le asignan roles dentro del sistema.
	ESTUDIANTES	Muestra todos los estudiantes registrados, pudiendo añadir nuevos, buscar y ver su perfil.
	PROFESORES	Muestra todos los profesores registrados, pudiendo añadir nuevos, buscar y ver su perfil.
	TRADUCTORES	Muestra todos los traductores registrados, pudiendo añadir nuevos, buscar y ver su perfil.
REGISTROS	CLASES	Permite añadir una nueva actividad seleccionando el profesor, estudiantes y el aula, editar el costo de la misma.
	DISPONIBILIDAD	Permite registrar un horario disponible para un profesor
	TRADUCCIONES	Registra y muestra el listado de las ultimas traducciones, subir al servidor el documento a traducir.
	CURSOS	Registra, lista los cursos que se imparten dentro del instituto.
	AULAS	Registra, lista las aulas dentro de la infraestructura del instituto.
	MONEDAS	Registra, lista las monedas y los tipos de cambio que se manejan en los pagos.
HORARIOS	PROFESORES	Genera horarios de profesores por semana, permitiendo mostrar los reportes realizados por los profesores y sus costos
	ESTUDIANTES	Genera horarios de estudiantes por semana
	AULAS	Genera horarios por aulas por semana
PAGOS	NUEVO	Permite generar y guardar y/o imprimir pagos a traductores, profesores y estudiantes.
	INFORME GASTOS	Despliega reporte por fechas de gastos que se incurren al pagar a profesores y/o traductores
	INFORME INGRESOS	Despliega reporte por fechas por ingresos que se incurren cuando se registran pagos los estudiantes

TABLA 3.23 DISEÑO NAVEGACIONAL DEL SISTEMA

### 3.4.3 MODELO DE RECURSOS DEL SISTEMA

Se planifica la arquitectura del servicio, mediante REST, esta es la planificación propuesta:

Esto simplificará urls, elevará la simplicidad de uso y además especificará el tipo de recursos que ofrecerá el sistema.

Url	Recurso	Acciones	Soporte
<b>Sección administrativa</b>			
/inscritos	Usuarios del sistema	Listar, añadir, modificar	html, xml, xls
inscritos/[id]/perfil	Perfil de un usuario	Mostrar, modificar	html, xml
/estudiantes	Estudiantes inscritos	Listar, añadir, modificar	html, xml, xls
/estudiantes/[id]/clases	Clases de un estudiante	Listar, mostrar horarios por semana, añadir, modificar	html, xml, js
/estudiantes/[id]/pagos	Pagos de los estudiantes	Listar, añadir, modificar	html, xml, xls
/profesores	Profesores registrados	Listar, añadir, modificar	html, xml, xls
/profesores/[id]/horarios	Horarios disponibles del profesor	Listar, mostrar horarios por semana, añadir, modificar	html, xml, js
/profesores/[id]/clases	clases registradas del profesor	Listar, mostrar horarios por semana, añadir, modificar	html, xml, js
/profesores/[id]/pagos	Pagos a profesores	Listar, reportar, calcular, modificar	html, xml, xls
/monedas	Monedas que se manejan en el sistema	Listar, añadir, modificar	html, xml
/monedas/[id]/cambios	Cambios de moneda	Listar, añadir, modificar	html, xml
/cursos	Cursos que se dictan en el instituto	Listar, añadir, modificar	html, xml
/aulas	Aulas que tiene el instituto	Listar, añadir, modificar	html, xml
<b>Sección profesores</b>			
/disponibles	Horarios disponibles del profesor	Listar, mostrar horarios por semana, añadir, modificar	html, xml, js
/clases	Clases del profesor	Listar, mostrar horarios por semana, añadir, modificar	html, xml
/students	Estudiantes del profesor	Listar	html, xml
/students/[id]/clases	Clases del estudiante	Mostrar horarios por semana, ver reportes	html, xml, js
/perfiles	Perfil de un usuario	Mostrar, modificar	html, xml
<b>Sección estudiantes</b>			
/lessons	Clases del estudiante	Listar, mostrar horarios por semana, añadir, modificar	html, xml
/perfiles	Perfil de un usuario	Mostrar, modificar	html, xml
<b>Sección traductores</b>			
/traducciones	Traducciones realizadas dentro del instituto	Listar, añadir, modificar, eliminar	html, xml, xls
/clientes	Datos del cliente de traducción	Listar, añadir, modificar, eliminar	html, xml

**Nota:** En el soporte si presenta js, tiene soporte para Ajax, si es xls es de Excel.

**TABLA 3.24** DEFINICIÓN DEL MODELO DE RECURSOS, REST

### 3.4.4 TÉCNICAS DE DESARROLLO DE SOFTWARE PROFESIONAL

A continuación se describen los conceptos para que el sistema se adentre en los conceptos de web 2.0 y desarrollo de software profesional.

#### 3.4.4.1 USO DE FRAMEWORKS DE APLICACIONES

Dentro de la etapa de investigación, se encuentra que el desarrollo de software, como parte de su evolución natural, se va centrando ahora en el uso de frameworks de aplicaciones, estas nos da un conjunto de funciones comunes desarrolladas para afrontar problemas comunes (manejo de datos, validaciones, etc..) facilitando el desarrollo y centrarnos totalmente en la solución y no los detalles.

Dentro del proyecto se utiliza los frameworks: Ruby on Rails y Prototype. El primero una plataforma de desarrollo web bajo el patrón MVC y el segundo un conjunto de funciones javascript para la utilización de la tecnología Ajax.

##### 3.4.4.1.1 PATRÓN DE ARQUITECTURA MVC

Todos los desarrolladores de software buscamos constantemente reutilizar código una y otra vez en todos los proyectos que vamos desarrollando. Cuando adquirimos experiencia, el sistema de programación cada vez se acerca más a un sistema modular, basado en capas o grupos. Esto ha dado lugar a los patrones de diseño, que no son más que guiones a seguir para llegar a un mismo resultado.

En cuanto al aspecto técnico de la aplicación, este modelo de arquitectura nos proporciona varias ventajas:

- Hay una clara separación entre los componentes de un programa; lo cual nos permite implementarlos por separado.

- Existe una interfaz muy bien definida entre las capas, que permite modificar o reemplazar el modelo, la vista o el controlador de forma transparente a las demás capas.
- La conexión entre el modelo y sus vistas es dinámica, es decir, se produce en tiempo de ejecución, no en tiempo de compilación.

Y desde el punto de vista científico, la elección del patrón modelo-vista-controlador para la implementación del presente proyecto se debe fundamentalmente a dos razones:

- De todos los patrones de diseño orientados a las aplicaciones Web, el patrón MVC es el más extendido, más documentado, y por lo tanto con mayor interés científico.
- Es independiente de las tecnologías utilizadas. El proyecto está desarrollado en Ruby on Rails, pero el conocimiento adquirido del modelo MVC es aplicable a cualquier lenguaje de programación o entorno de desarrollo de aplicaciones Web.

#### 3.4.4.1.2 AJAX

El uso de la tecnología AJAX en la aplicación está destinado a agilizar los tiempos de respuesta y la interactividad de aquellas áreas que más necesitan de esta interactividad, un ejemplo claro es el registro de clases de un profesor, dentro de la manera tradicional se tardaría el triple de tiempo el registrar clases sumando que pasan muchas clases por día y no siempre existen maquinas dentro del instituto lo cual fomenta la no utilización del sistema y la desacreditación del mismo.

Las estructuras que permiten el uso de esta tecnología en la aplicación se dividen entre el lado del cliente, con el motor JavaScript AJAX, y los controladores del lado del servidor que dan cobertura a las peticiones del lado del cliente.

##### **Lado del cliente**

Aquí tenemos por un lado el motor AJAX general, empotrado al navegador de internet, y el framework Prototype que permite la actualización in-situ de la pagina como un ejemplo está

el de adicionar una clase dentro de la semana que se muestra en la pagina y esta se actualiza solamente en el día que se añadió la clase.

El framework Prototype permite realizar las siguientes acciones dentro del lado del cliente:

- Envío de peticiones y datos al servidor.
- Recepción de datos y presentación de los mismos en el área de la página especificada.
- Ejecución de código de respuesta personalizado.
- Visualización de icono de carga.
- Serialización de los campos de un formulario para su inclusión en peticiones al servidor.

#### **Lado del servidor**

En cuanto al lado del servidor, conjunto de clases encapsuladas dentro del framework Ruby on Rails, las que reciben y dan servicio a las peticiones del motor AJAX del cliente. El proceso de atención de una petición, de forma general, es el siguiente:

- Se recibe la petición procedente del motor AJAX del cliente y se validan los datos de la misma.
- Se efectúan las operaciones necesarias, interactuando con las clases del modelo.
- Se realiza un filtraje del tipo de recurso que se desea (HTML, XML, JS, etc.) si es javascript (Ajax) se responde con un archivo [recurso].js y este es consumido por el cliente.
- Se ejecuta el parser de javascript dinamico de Rails, los RJS.
- El resultado generado es enviado al motor AJAX, que se ha mantenido a la espera, y este trata el código devuelto lo que permite la actualización de la página que se muestra en el navegador del cliente.

#### **3.4.4.1.3 MANEJO DE VERSIONADO DE CÓDIGO FUENTE**

Dentro del proceso de desarrollo de software es imprescindible la utilización de versionado del código fuente, que nos permiten acciones del tipo: control de cambios, pérdida de

información, etc. Por este motivo se utiliza subversion (sección 2.7) y se configura un servidor de versiones.

### 3.4.5 GESTIÓN DEL PROYECTO

Como se escribió anteriormente se maneja la metodología de gestión SCRUM, como un complemento a RUP.

#### 3.4.5.1 DEFINICIÓN DE ROLES DEL SCRUM

Product Owner: Sra. Anne Marie Van Swinderen.

Scrum Master: Carlos Alberto Ramos.

#### 3.4.5.2 DESARROLLO DE LOS SPRINTS

Antes de cualquier sprint se debe de estimar, organizar y repartir trabajos del sprint como tal, para ello se realiza el Sprint Backlog, donde se definirá la funcionalidad que se presentará y la duración del sprint.

##### 3.4.5.2.1 PRIMER SPRINT

Se reparten las tareas que se vean convenientes, realizando la estimación de la tarea y la preparación del ejecutor de la tarea.

Item	Tareas	Tipo	Estado	Responsable	Est.
1	Elaboración de la autenticación del sitio	código	asignado	Carlos	15
2	Elaboración y refinación de las tablas de datos	código	asignado	Carlos	5
3	Permitir a profesores registrar sus clases sin choques de horario	código	asignado	Carlos	20
4	Permitir a los profesores registrar sus horarios disponibles	código	asignado	Carlos	20
5	Permitir registrar aulas solo en la parte administrativa	código	asignado	Carlos	5
6	Controlar permisos y roles de usuario, a lo largo del sitio	código	asignado	Carlos	15
7	Implementar que cada usuario pueda rellenar su perfil	código	asignado	Carlos	5
8	Elaborar manejo de usuario, estudiantes,	código	asignado	Carlos	10

	profesores y administradores				
9	Permitir añadir monedas y manejar los cambios de moneda	código	asignado	Carlos	5
10	Diseño del sitio	diseño	asignado	Carlos	5
11	Conseguir nuevo hosting, con soporte de Ruby on Rails	servidor	asignado	Carlos	20
12	Configuración del los correos electrónicos del dominio	servidor	asignado	Carlos	10
<b>Horas</b>					<b>135</b>

**TABLA 3.25 PRIMER SPRINT BACKLOG**

Tenemos como el Story Point o la estimación del esfuerzo para el primer sprint es de:

135 horas, el cual podemos convertir a días, con la relación de:

<b>6 HORAS ESFUERZO = 1 DÍA DE ESFUERZO (ECUACIÓN 3.1)</b>
--

135 horas = 23 días, pero en la práctica, la metodología sugiere tener para el primer sprint una duración de un mes, el cual es aceptado por el Product Owner.

La culminación del primer sprint es dentro de un mes.

### 3.4.5.2.2 SEGUNDO SPRINT

Se reparten las tareas que se vean convenientes, realizando la estimación de la tarea y la preparación del ejecutor de la tarea.

Ítem	Tareas	Tipo	Estado	Resp.	Est.
1	Permitir a profesores rellenar un informe pasada la clase con un estudiante.	código	asignado	Carlos	5
2	Permitir registrar en cada clase más de un alumno.	código	asignado	Carlos	10
3	Permitir que puedan añadir una clase mediante llamadas ajax.	código	asignado	Carlos	10
4	Permitir que se pueda eliminar una clase mediante llamadas ajax.	código	asignado	Carlos	10
5	Preparar el sitio para poder traducirlo al ingles.	diseño	asignado	Carlos	20
6	Colocar un calendario para seleccionar la fecha, tanto en el registro de clases como en el registro de horarios disponibles.	código	asignado	Carlos	5
7	Permitir a los profesores que puedan ver las clases de sus alumnos, compartiendo los informes generados por ellos mismos.	código	asignado	Carlos	9

8	Paginar los listados de usuarios, estudiantes y profesores.	código	asignado	Carlos	5
9	Manejar permitir añadir, quitar, modificar a administradores, el crédito de los estudiantes.	código	asignado	Carlos	5
10	Mostrar si un profesores esta contratado o no, además asignarle un porcentaje de ganancia.	código	asignado	Carlos	3
11	Configurar las cuentas de correo electrónico.	servidor	asignado	Carlos	10
12	Elaboración de los manuales de uso.	manual	asignado	Carlos	3
13	Permitir a estudiantes ver los horarios disponibles de profesores, mediante ajax, controlando su crédito.	código	asignado	Carlos	10
<b>Horas</b>					<b>105</b>

**TABLA 3.26 SPRINT BACKLOG DEL SEGUNDO SPRINT**

La duración del segundo sprint se define dentro de tres semanas.

### 3.4.5.2.3 TERCER SPRINT

Se reparten las tareas que se vean convenientes, realizando la estimación de la tarea y la preparación del ejecutor de la tarea.

Item	Tareas	Tipo	Estado	Resp.	Est.
1	Añadir de google maps la ubicación del instituto.	código	asignado	Carlos	5
2	La traducción del sitio en inglés.	diseño	asignado	Carlos	20
3	Permitir añadir distintas clases en una sola acción, máximo cinco.	código	asignado	Carlos	5
4	Permitir enviar a los estudiantes un correo electrónico.	código	asignado	Carlos	15
5	Diseñar e implementar los pagos de los profesores.	código	asignado	Carlos	15
6	Permitir registrar las traducciones.	código	asignado	Carlos	5
7	Registrar a clientes de traducciones.	código	asignado	Carlos	5
8	Implementar el llenado de perfiles.	código	asignado	Carlos	7
9	Permitir que se puedan subir archivos al sitio.	código	asignado	Carlos	10
10	Añadir seleccionar idiomas de traducción	código	asignado	Carlos	5
11	Rediseñar el sitio	diseño	asignado	Carlos	25
<b>Horas</b>					<b>117</b>

**TABLA 3.27 SPRINT BACKLOG DEL TERCER SPRINT**

La duración del tercer sprint se define dentro de tres semanas.

El total estimado del tiempo de desarrollo del proyecto es de 2 meses y dos semanas.

### 3.4.5.2.4 DESARROLLO DEL PORTAL EN CADA ITERACIÓN O SPRINT

La forma de mostrar la funcionalidad añadida dentro de cada sprint, se muestran partes del sitio ya funcionales:



FIGURA 3.10 FUNCIONALIDAD DENTRO DEL PRIMER SPRINT, MOSTRAR HORARIOS POR SEMANA (PROFESOR)



FIGURA 3.11 FUNCIONALIDAD DENTRO DEL SEGUNDO SPRINT, SOPORTE DE AJAX

## Instituto Exclusivo de Idiomas

!!! Bienvenidos !!!

Enseñanza de español a todo el mundo a través de Internet...

Español on-line.



### Enlaces

- Instituto exclusivo de Idiomas
- Materiales del instituto exclusivo
- Desarrollo de sitios web 2.0

FIGURA 3.12 FUNCIONALIDAD DENTRO DEL TERCER SPRINT, REDISEÑO DEL SITIO

Logged in as Iris Logout ◀ Go to home English Español

Schedules Classes Students Courses My profile Instituto Exclusivo **i.e.** beta

### Class schedule from: Iris Register many classes | New class | List

← Time in the institute: 2007-12-11 23:26:01 →

	10/12/2007 Monday	11/12/2007 Tuesday	12/12/2007 Wednesday	13/12/2007 Thursday	14/12/2007 Friday	15/12/2007 Saturday	16/12/2007 Sunday
08:00		08:00 to 09:30 Adam Ranson	08:00	08:00	08:00	08:00	08:00
08:30			08:30	08:30	08:30	08:30	08:30
09:00			09:00	09:00	09:00	09:00	09:00
09:30		09:30	09:30	09:30	09:30	09:30	09:30
10:00		10:00	10:00	10:00	10:00	10:00	10:00

Select student: Adam Ranson Room: A Send me an email?:  Send an email to the student?:

Date: 10/12/2007 From: 09:00 To: 10:00

Create class

FIGURA 3.13 FUNCIONALIDAD DENTRO DEL TERCER SPRINT, AÑADIR UNA NUEVA CLASE MEDIANTE AJAX Y SOPORTE MULTILINGÜE (INTERFAZ PROFESOR)

### 3.4.6 MODELO DE PRUEBAS

#### 3.4.6.1 MODELO DE PRUEBAS UNITARIAS

El modelo de pruebas unitarias se realiza en todo momento dentro del desarrollo del proyecto, permite automatizar las pruebas del software. El modelo de las pruebas unitarias es de agarrar cada clase, crear instancias de la clase, someterlas a sus métodos y comparar los resultados.

Este es un ejemplo, prueba unitaria para la clase: Clase.

```
def test_adicionar_clases
  # Registramos una clase
  nueva_clase = Clase.new(:profesor_id => 1, :fecha => '2007-10-11', :hora_inicio
=> "20:00:00", :hora_fin => "21:00:00")
  assert_equal nueva_clase.save, true
  # Registramos una clase pero tiene un choque con la anterior
  otra_clase = Clase.new(:profesor_id => 1, :fecha => '2007-10-11', :hora_inicio
=> "19:30:00", :hora_fin => "20:30:00")
  assert_equal otra_clase.save, false
end
```

**TABLA 3.28** EJEMPLO DE UNA PRUEBA UNITARIA EN LENGUAJE DE PROGRAMACION RUBY

Se han realizado pruebas a todas las clases dentro del sistema.

### 3.7 CALIDAD DEL SOFTWARE

#### 3.7.1 MÉTRICA BASADA EN EL MODELO DE Mc CALL (ISO 9126)

La tabla nos muestra los resultados arrojados al aplicar la metodología descrita en la sección [], donde a cada factor de calidad se identifica criterios y se pondera una calificación dentro del rango de 0 a 10 el cual fue calificado por el personal dentro del instituto mediante un formulario.

Factor de calidad	Criterio	Ponderación	%
Funcionalidad	Idoneidad	8	88
	Corrección	9	
	Interoperabilidad	8	

	Conformidad	10	
	Seguridad	9	
<b>Confiabilidad</b>	Madurez	8	<b>83</b>
	Tolerancia a fallos	7	
	Facilidad de recuperación	10	
<b>Usabilidad</b>	Precisión	9	<b>86</b>
	Consistencia	9	
	Tolerancia a fallos	7	
	Modularidad	9	
	Simplicidad	9	
<b>Eficiencia</b>	Tiempo de uso	10	<b>90</b>
	Recursos utilizados	8	
<b>Facilidad de mantenimiento</b>	Facilidad de análisis	9	<b>87</b>
	Facilidad de cambio	8	
	Estabilidad	9	
	Facilidad de prueba	9	
<b>Portabilidad</b>	Facilidad de instalación	7	<b>83</b>
	Facilidad de ajuste	9	
	Facilidad de adopción al cambio	9	

**TABLA 3.29** RESULTADO DE METRICA DE MC CALL

Para realizar el cálculo del índice de calidad del producto, se hace uso de la ecuación, especificada en la sección 2.2

$$QT = (88 + 83 + 86 + 90 + 87 + 83) / 6 = 86\%$$

La calidad del software considerando los criterios y tomando en cuenta los factores de ponderación, da como resultado de grado de calidad del 86%, indica este factor un grado alto de calidad.

### 3.7.2 PUNTO FUNCIÓN

Para calcular la calidad por esta métrica, se tiene que dividir el sistema en módulos de esta manera el promedio del todo establecerá el nivel de calidad de todo el sistema.

## Actividades

	Conteo	Factor	Total
EE	4	3	12
SE	3	4	12
CE	2	3	6
ALI	3	7	21
AIE	3	5	15
<b>Conteo total:</b>			<b>66</b>

## Pagos

	Conteo	Factor	Total
EE	2	3	6
SE	2	4	8
CE	2	3	6
ALI	4	7	28
AIE	2	5	10
<b>Conteo total:</b>			<b>58</b>

## Traducciones

	Conteo	Factor	Total
EE	5	3	15
SE	2	4	8
CE	2	3	6
ALI	2	7	14
AIE	3	5	15
<b>Conteo total:</b>			<b>58</b>

**TABLA 3.30** CALCULO DE PUNTOS DE CARACTERISTICAS

Nº	Preguntas	Actividades	Pagos	Traducciones
1	¿Requiere el sistema copias de seguridad y de recuperación fiables?	2	2	1
2	¿Se requiere de comunicación de datos?	3	1	1
3	¿Existen funciones de procesamiento distribuido?	0	0	0
4	¿Es crítico el rendimiento?	0	0	0
5	¿Se ejecutara el sistema en un entorno operativo existente y frecuentemente utilizado?	0	0	0
6	¿El sistema requiere entradas de datos interactiva?	1	1	1
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples o variadas operaciones?	1	0	0
8	¿Se actualizan los archivos maestros en forma interactiva?	0	0	0
9	¿Son complejas las entradas, salidas, archivos o peticiones?	0	0	0
10	¿Es complejo el procesamiento interno?	1	2	1
11	¿Se ha diseñado el código para ser reutilizado?	1	0	0
12	¿Están incluidas en el diseño la conversión y la instalación?	0	0	0
13	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	1	0	0
14	¿Se ha diseñado el sistema para soportar múltiples	1	1	1

instalaciones en diferentes organizaciones?			
<b>SUM (Fi)</b>	<b>11</b>	<b>7</b>	<b>5</b>

**TABLA 3.31** CALCULO DEL FACTOR DE AJUSTE

Utilizando las ecuaciones 2.3 y 2.4, tenemos los siguientes datos:

	<b>Actividades</b>	<b>Pagos</b>	<b>Traducciones</b>
<b>PF</b>	56	78	56
<b>Errores</b>	0.12	0.12	0.12
<b>Calidad</b>	0.771	0.756	0.672

**TABLA 3.32** CALIDAD UTILIZANDO PUNTO FUNCION

Con el resultado encontrado en la tabla 3.21, podemos aplicar el promedio de los módulos para así poder conseguir la calidad de software:

$$PF = 77\% + 76\% + 67\%$$

La calidad evaluada por el punto función nos da un índice de calidad del software del **73%**.

### 3.7.3 USABILIDAD CON AJAX

De acuerdo al anexo B, se han hecho las siguientes mejoras dentro de la interfaz del sitio:

- Ante cambios tenues del interface, se incorporan señales sutiles para que el usuario perciba que algo ha cambiado en la página.
- Se aporta información adicional sobre el estado del sistema, una caja de texto que muestra información sobre el estado del sistema.
- El uso de colores para denotar la interacción como pulsaciones que permiten mostrar los cambios dentro del sistema.
- Con todo esto si el usuario no cuenta o tiene el java script deshabilitado, el sistema todavía puede realizar acciones dentro del sistema.

### 3.8 SEGURIDAD DEL SOFTWARE

En esta sección se evaluara los posibles riesgos que puedan afectar al sistema.

#	Amenaza	D	R	E	A	D	Σ
---	---------	---	---	---	---	---	---

1	Inyección de SQL	2	1	1	2	1	<b>7</b>
2	Script entre sitios	1	2	1	3	2	<b>9</b>
3	Modificación del ingreso.	1	2	2	2	2	<b>9</b>
4	Robo de claves.	1	2	1	3	1	<b>8</b>
5	Reemplazo de sesión.	1	2	1	1	1	<b>6</b>
6	Reemplazo de identidad.	3	1	2	1	1	<b>8</b>
7	Revelado de información	2	2	2	3	1	<b>10</b>

**TABLA 3.33** EVALUACION DE LAS VULNERABILIDADES SEGUN DREAD

En la tabla 3.28 se puede observar que las filas script revelado de información, script entre sitios y la modificación del ingreso, son los que generan un valor más alto y deben de ser consideradas en todo el ciclo de vida del software, además el framework Ruby on rails ya tiene funciones desarrolladas para la seguridad, se los está usando y además se actualiza cada cierto tiempo contra nuevas vulnerabilidades de seguridad.

Para eliminar la inyección SQL se utilizan conversiones seguras, que preformatean las entradas para que no entre código inseguro.

Además en campos requeridos, de tipo moneda y fechas se utilizan funciones javascript que evalúan las entradas y muestran los errores que ingreso el usuario.

Para eliminar el Script entre sitios, para el ingreso y despliegue de datos desde la base de datos, los mismos son formateados para que no sean dañinos.

Para eliminar el reemplazo de sesión se da un tiempo de vida de la sesión, en este si no existe actividad dentro de 30 minutos la sesión expira, pidiendo de nuevo que se autentifique el usuario imposibilitando el robo de sesiones, además se utiliza para la sección administrativa el registro de la ip, bloqueando el acceso desde otro lugar distinto del instituto.

# Capítulo IV

## 4.1 CONCLUSIONES

Con la implementación del “Sistema Web integrado de planificación pedagógica y pagos”, para el Instituto Exclusivo de Idiomas, se puede observar que:

- ✓ Manejo automatizado de usuarios, asignando roles automáticamente.
- ✓ Inscripción de estudiantes y reservar una clase mediante el sistema a través del internet, con una interfaz multilingüe.
- ✓ Generación de horarios para administración y profesores.
- ✓ Comunicación más acorde con la tecnología actual.
- ✓ Gestión de crédito y clases de estudiantes.
- ✓ Pago a profesores mediante reglas dentro del instituto.
- ✓ Registro de traducciones y sus pagos.
- ✓ Informes estadísticos pertinentes.

Todo esto logra cumplir en su totalidad: Implementado el sistema web integrado de planificación pedagógica y control de pagos, que permite el registro de usuarios, gestión y coordinación de actividades pedagógicas como las clases de español optimizando el manejo de recursos y el control de pagos de estudiantes y profesores dentro del Instituto Exclusivo de Idiomas aplicando la tecnología actual y el concepto de web 2.0.

Gracias a la plataforma de trabajo, Ruby on Rails, se pudo realizar el trabajo dentro de los parámetros de la Web 2.0, integración de la tecnología Ajax, diseño separado de los datos, comunicación, etc.

Además se adentro al manejo de nuevos enfoques de metodología y tecnología:

Lenguaje de programación: **Ruby**, Plataforma de trabajo: **Rails.**, Plataforma de trabajo Ajax: **Prototype**, Servidor web: **Lighttpd**, Tecnología: **Ajax**, Arquitectura del servicio: **Rest**, Manejo de versionado del código fuente: **Subversion**, Asimilación del concepto de **web 2.0**.

## 4.2 RECOMENDACIONES

Se recomienda que se realice el diseño del modulo de pagos a través de tarjeta de crédito, para permitir a personas del resto del mundo, puedan reservar clases.

Además se recomienda la generación de facturas digitales de acuerdo a normas de impuestos, que entraran en vigencia en el 2008.

El análisis realizado para el instituto durante la primera fase de desarrollo del sistema, dio pautas para sugerir los siguientes proyectos:

- ✓ Elaboración de sistema contable que controle e integre los pagos que se realizan a través del sistema.
- ✓ Sistema de catalogación de material de enseñanza, El Instituto maneja un volumen considerable de material bibliográfico, el cual no está a disposición total de profesores y estudiantes.
- ✓ Sistema experto para la evaluación del nivel de aprendizaje de idiomas.

## REFERENCIA BIBLIOGRÁFICA:

[PRE98] PRESSMAN Roger, "INGENIERÍA DE SOFTWARE, un enfoque practico" (cuarta edición), Editorial. McGraw-Hill Inc., España, 580 pág.

[JBR99] Jacobson Ivan, Grady Booch, Rumbaugh James, "El proceso Unificado de Desarrollo de Software", 1999. España: Pearson Educacion S.A.

[CRP07] Cobo Romaní Cristobal y Pardo Kuklinski Hugo, "PLANETA WEB 2.0. Inteligencia colectiva o medios fast food", Grup de Recerca d'Interaccions Digitals, Universitat de Vic. Flacso México. Barcelona / México DF, 162 pág. [PDF].

[FUR07] Fumero Antonio y Roca Genís, "WEB 2.0", Fundación Orange, España, 131 pág. [PDF].

[DHH06] Dave Thomas y David Heinemeier Hansson, "AGILE WEB DEVELOPMENT WITH RAILS", (segunda edicion) editorial The Pragmatic Programmers 2006.

[WRB07] Wirdemann Ralf y Baustert Thomas, "DESARROLLO REST CON RAILS", B-simple, Alemania – Hamburgo, 41 pág. [PDF]

[KNI07] HENRIK Kniberg, "SCRUM AND XP FROM THE TRENCHES", 2007, Estados Unidos, Editorial. C4Media.

[PAL07] Palacio Juan, "Flexibilidad con scrum", 2007, España, 179 pag.

### DOCUMENTOS WEB:

[WIK07] **Wikipedia** Enciclopedia libre del idioma español, Septiembre 2007  
<http://www.es.wikipedia.org>.

[AJA07] Manual para aprender sobre la tecnología Ajax, Agosto 2007,  
<http://www.ajaxya.com.ar>

**[SUB07]** Control de versiones con Subversion, noviembre 2007,  
<http://www.subversion.tigris.org>

**[ROR07]** Sitio Oficial del framework Ruby on rails, octubre 2007, <http://www.rubyonrails.org>

**[SOFT07]** "SCRUM IN FIVE MINUTES", septiembre 2007,  
[www.softhouse.se/Uploades/Scrum\\_eng\\_webb.pdf](http://www.softhouse.se/Uploades/Scrum_eng_webb.pdf)

#### **BLOGS:**

**[COR07]** Corral Rodrigo - La masa, el ladrillo, la bota, el bocadillo...,  
<http://geeks.ms/blogs/rcorral/default.aspx>

**[GOZ07]** Gozalves Vicent, Mi ventana web 2.0, <http://www.vicentgozalves.com>

**[NAV07]** Blog sobre metodologías de desarrollo de Software, <http://www.navegapolis.net/>



## GLOSARIO DE TÉRMINOS

**Ajax no obstrusivo.**- Java Script de tipo no obstructivo, Ajax necesita que este habilitado la ejecución de java script, si no está habilitado la página no realiza nada, se cuelga.

**ECMAScript.**- ECMAScript es una especificación de lenguaje de programación publicada por ECMA International. El desarrollo empezó en 1996 y estuvo basado en el popular lenguaje JavaScript propuesto como estándar por Netscape. Actualmente está aceptado como el estándar ISO 16262.

**Jscript.** JScript es la implementación de Microsoft de ECMAScript. Está disponible mediante Internet Explorer y el Windows Scripting Host. La versión más reciente es JScript .NET, que está basado en la versión 4 del estándar ECMAScript (aún no terminado), y puede ser compilado para la plataforma Microsoft .NET.

**XHttpRequest.**- Objeto dentro de un navegador cliente que permite realizar llamadas asincrónicas al servidor, componente principal para las llamadas de tipo Ajax.

**IP:** Protocolo del Internet, estándar o formato para conexiones de ordenadores con la red del Internet

**Prototype.**- Prototype es un framework basado en JavaScript que se orienta al desarrollo sencillo y dinámico de aplicaciones web. Es una herramienta que implementa las técnicas AJAX y su potencial es aprovechado al máximo cuando se desarrolla con Ruby On Rails.

**Lighttpd.**- lighttpd es un servidor web diseñado para ser rápido, seguro, flexible, y fiel a los estándares. Está optimizado para entornos donde la velocidad es muy importante, y por eso consume menos CPU y memoria RAM que otros servidores. Por todo lo que ofrece, lighttpd es apropiado para cualquier servidor que tenga problemas de carga.

**Virtual Private Server.**- Virtualización de hardware por medio de software, permite emular varios servidores dentro de un servidor.

## ANEXO A – SCRUM: MODELO DE DESARROLLO ÁGIL DE PRODUCTOS

Aunque surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software.

Jeff Sutherland aplicó el modelo Scrum al desarrollo de software en 1993 en Easel Corporation (Empresa que en los macro-juegos de compras y fusiones se integraría en VMARK, luego en Informix y finalmente en Ascential Software Corporation). En 1996 lo presentó junto con Ken Schwaber como proceso formal, también para gestión del desarrollo de software en OOPSLA 96. Más tarde, en 2001 serían dos de los promulgadores del Manifiesto ágil. En el desarrollo de software scrum está considerado como modelo ágil por la Agile Alliance.

### A.1 ROLES Y ACTORES

Para tener una idea clara de la filosofía del SCRUM, tenemos una interesante historia:

	Una gallina y un cerdo paseaban por la carretera.	
<p>La gallina dijo al cerdo: “Quieres abrir un restaurante conmigo”. El cerdo consideró la propuesta y respondió: “Sí, me gustaría. ¿Y cómo lo llamaríamos?”. La gallina respondió: “Huevos con tocino”.</p>		
<p>El cerdo se detuvo, hizo una pausa y contestó: “Pensándolo mejor, creo que no voy a abrir un restaurante contigo. Yo estaría realmente comprometido, mientras que tu estarías sólo implicada”.</p>		

**TABLA A.1** HISTORIA DE LA GALLINA Y EL CERDO

**COMPROMETIDOS EN EL PROYECTO.**- Tienen autoridad para hacer lo necesario para que salga adelante el proyecto. Están totalmente comprometidos con el proyecto para lograr el éxito:

- Dueño del producto (Product Owner).
- Equipo (Scrum Team).
- Scrum Master.

**IMPLICADOS EN EL PROYECTO.**- No deben interferir innecesariamente. Pueden estar interesados en el éxito del proyecto, pero no tienen autoridad directa sobre el progreso y la ejecución del mismo:

- Departamento de Marketing.
- Departamento Comercial.
- Usuarios o Clientes.

## A.2 DEFINICIÓN DE ROLES:

- El **Product Owner** conoce y marca las prioridades del proyecto o producto.
- El **Scrum Master** es la persona que asegura el seguimiento de la metodología guiando las reuniones y ayudando al equipo ante cualquier problema que pueda aparecer. Su responsabilidad es entre otras, la de hacer de paraguas ante las presiones externas.
- El **Scrum Team** son las personas responsables de implementar la funcionalidad o funcionalidades elegidas por el Product Owner.
- Los **Usuarios o Cliente**, son los beneficiarios finales del producto, y son quienes viendo los progresos, pueden aportar ideas, sugerencias o necesidades.

### A.3 FLUJO DE TRABAJO

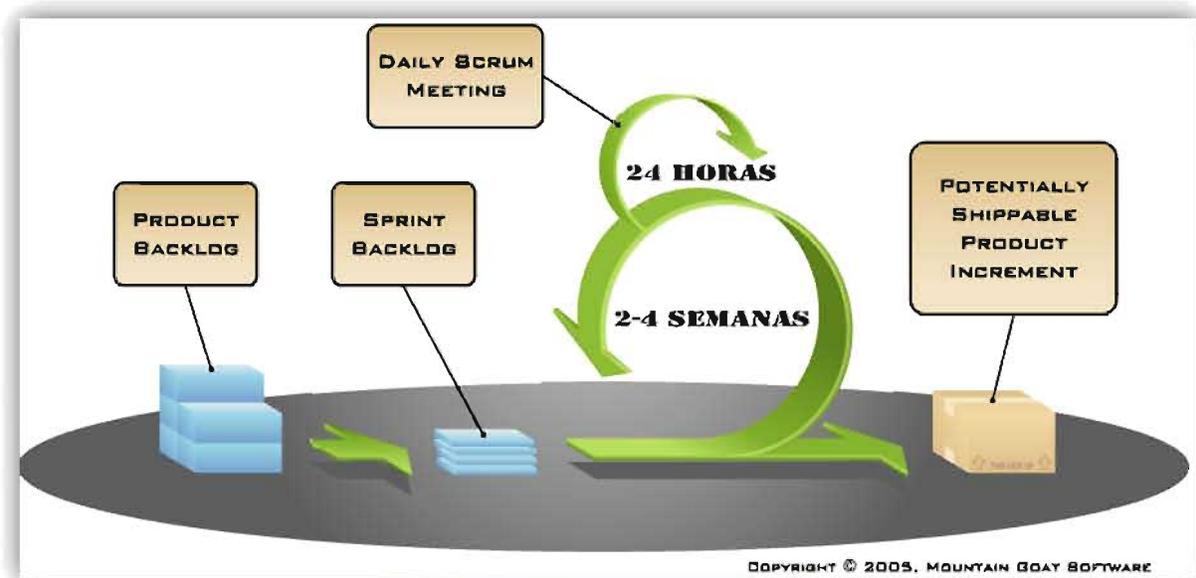


FIGURA A.0.1 CICLO DE DESARROLLO DENTRO DE SCRUM  
FUENTE: SCRUM IN FIVE MINUTES

La figura hace descripción sinóptica del proceso y sus elementos, que son:

- Product Backlog
- Sprint Backlog
- Daily Scrum Meeting

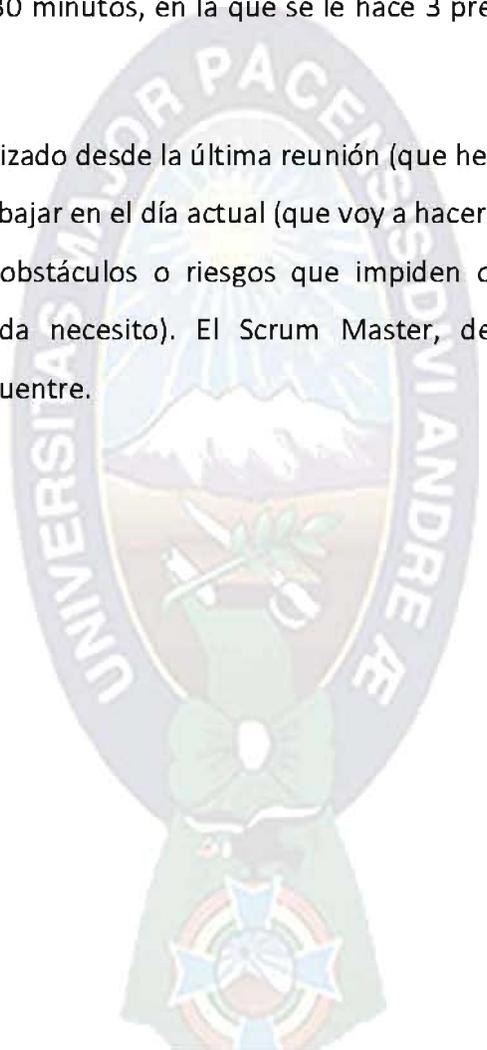
El **Product Backlog** corresponde con todas las tareas, funcionalidades o requerimientos a realizar. El Product Owner es la persona que se encarga de marcar las prioridades, y es al fin y al cabo, la persona que mantiene y actualiza dado el caso, la lista de tareas.

El **Sprint Backlog** corresponde con una o más tareas que provienen del Product Backlog. Es decir, del Product Backlog se saca una o más tareas que van a formar parte del Sprint Backlog. Las tareas del Sprint Backlog se deben acometer (recomendado) en unas 2 semanas ó 4 semanas. Hay Sprint Backlogs de 2 semanas y hay Sprint Backlogs de 4 semanas. Eso debe de ser marcado antes de iniciar el Sprint Backlog, de hecho, del Product Backlog se sacará la tarea o tareas realistas para acometer el Sprint Backlog. Una norma fundamental

es que mientras un Sprint Backlog se inicia, éste NO puede ser alterado o modificado. Hay que esperar a que concluya el Sprint Backlog para realizar la correspondiente modificación o alteración cuya tarea, formaría parte de otro Sprint Backlog.

El **Daily Scrum Meeting** es una tarea iterativa que se realiza todos los días que dure el Sprint Backlog con el equipo de desarrollo o de trabajo. Se trata de una reunión operativa, informal y ágil, de un máximo de 30 minutos, en la que se le hace 3 preguntas a cada integrante del equipo.

- ✓ Qué tareas ha realizado desde la última reunión (que he hecho).
- ✓ Sobre qué va a trabajar en el día actual (que voy a hacer hoy).
- ✓ Identificación de obstáculos o riesgos que impiden o pueden impedir el normal avance (que ayuda necesito). El Scrum Master, debe eliminar aquí cualquier obstáculo que encuentre.



## ANEXO B - AJAX Y LA USABILIDAD

La principal virtud de AJAX está en la potencia que se le puede extraer al trabajo asíncrono de peticiones al servidor. Estamos acostumbrados a un modelo de interacción sincrónica basada en clic-petición-presentación, clic-petición-presentación. Con AJAX la interacción pasa a ser asíncrona. Cada vez que se hace clic no necesariamente se establece una conexión con el servidor.

Ahí está el meollo precisamente, en que ciertos procesos se muestran en la página sin retardo alguno, pues mientras el usuario miraba unos datos en la pantalla, AJAX le ha estado preparando los siguientes que iba a necesitar.

AJAX se utiliza para algunas de las últimas aplicaciones de Google, cuyos desarrolladores parecen haber entendido a la perfección en qué consiste Internet y sus potencialidades.

Por ejemplo, si he entendido bien el mecanismo de funcionamiento asíncrono de AJAX, cuando leemos el correo en Gmail y abrimos un mensaje que forma parte de una cadena de mensajes con alguien (o conversación) sólo se nos muestra el último mensaje recibido del emisor. Mientras nosotros leemos ese mensaje Gmail va cargando el resto de mensajes de esa conversación, de forma que cuando pulsamos en la opción de Expandir los mensajes vemos que se carga inmediatamente y que la url no varía.

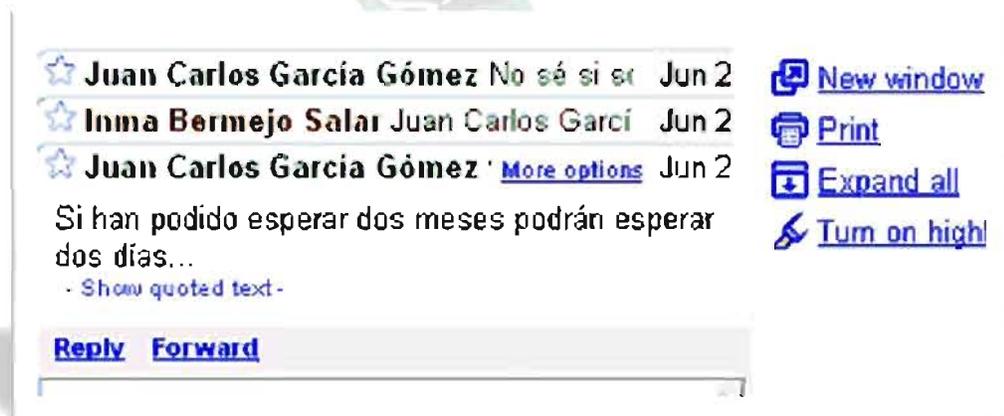


FIGURA A.0.1 APLICACIÓN WEB QUE UTILIZA AJAX (GMAIL)

FUENTE: GMAIL.COM

También se utiliza AJAX para Google Maps y para Google Suggest.

Las vías del desarrollo futuro aún no está claro hacia dónde puede derivar, pues esto no ha hecho más que empezar. Por ejemplo ahora los desarrolladores ya se han percatado de las posibilidades que ofrece modificar una simple aplicación para hacer peticiones-ajax en el web.

En realidad la tecnología en sí no es especialmente nueva, ni siquiera la idea del trabajo asíncrono (Flash y Java ya lo permitían), pero sí tal vez las vías de aplicación y los resultados en cuanto a experiencia de usuario. Ahí, claro está, también hay sus problemas.

## B.1 PROBLEMAS

Estas aplicaciones tienen una pinta estupenda, y resultan tremendamente útiles. Los problemas, sin embargo, vienen precisamente de la mano de la propia tecnología empleada.

El hecho de emplear javascript y XMLHttpRequest ya implica problemas con los navegadores que no los soporten o no los tengan habilitados.

La solución, obviamente, pasa por ofrecer una alternativa sin usar esa tecnología, pero claro, en ese caso perdería parte del interés que tiene una aplicación como Google Maps, ya que, dada la cantidad de datos que ha de manejar, sería muy poco útil si tuviera que trabajar con el clásico método sincrónico. El tiempo de espera sería, tal vez, demasiado largo como para que al usuario le valiera la pena esperar la llegada de los resultados.

De otro lado, están los problemas relacionados con la experiencia de usuario en sí misma:

- 1) Los formularios basados en XMLHttpRequest no actúan como el usuario está acostumbrado. En un formulario clásico uno puede hacer y deshacer a placer, sabiendo que hasta que no se pulse el botón “enviar” podemos cambiar los datos que haga falta. Pero aquí no funciona así, los datos se envían campo a campo, lo que desvirtúa el concepto clásico de formulario y se introduce un elemento que genera gran perturbación y confunde al usuario más avanzado.

- 2) La carga instantánea de una nueva página (que en realidad es la misma) puede producir en el usuario la sensación de que nada ha pasado, pues la carga ha sido muy rápida como para que pudiera percibirla y, además, la url no ha variado.
- 3) Si ya el sitio ya esta incorporado AJAX dentro de la interfaz, se debe de tener mucho cuidado que el usuario habilite o deshabilite la ejecución de javascript, parte fundamental de acciones AJAX.

Soluciones a los anteriores dilemas podrían ser las siguientes:

- Mediante AJAX (peticiones XMLHttpRequest) la recarga es parcial y la percepción del cambio es mucho menos notoria. De ahí que ante cambios tenues del interface, sea necesario incorporar señales sutiles para que el usuario perciba que algo ha cambiado en la página.
- Esto nos lleva a las cuestiones relacionadas con la visibilidad del estado del sistema. Es posible que de tan rápido que funciona la actualización de datos en pantalla el usuario no se percate del cambio y, por tanto, espere indefinidamente un cambio que ya se ha producido. Por tanto se debe aportar información adicional sobre el estado del sistema.
- El uso de colores para denotar la interacción deberá contar con el contraste suficiente para que el cambio sea significativo. Y, añadido, debería ser complementado con algún otro signo que invalidara esa técnica desde el punto de vista de la accesibilidad (Pauta 2: No se base sólo en el color).
- Con todo lo anterior sobresale un principio más simple: retardar ligeramente, de forma artificial, la carga de datos en la página, para así ofrecer la habitual sensación de página que se descarga poco a poco, y no de golpe como hace AJAX.
- La cuestión sería que el retraso fuera lo suficientemente leve como para no ralentizar el sistema y lo suficientemente largo como para que el usuario percibiera correctamente el cambio en el estado del sistema (la sensación de descarga de la página).
- Posibilitar la ejecución normal de la página a pesar de inhabilitar javascript.

## ANEXO C - PRODUCT BACKLOG FINAL

Product Backlog realizado entre el Product Owner (Dueño del sistema), y el Scrum Master.

#	Tarea	Est. dias	Descripción
<b>Prioridad - Muy Alto Nivel</b>			
1	Conseguir que las reservación de clases por usuarios sea posible	7	Cada profesor debe de registrar sus horarios disponibles con esta información los estudiantes podrán reservar una clase en estos horarios, viendo que también no tenga clases en esos horarios.
2	El cálculo de pagos por comisiones a profesores.	7	Los profesores a partir de sus clases registradas deben de llenar un pequeño informe, además el sistema debe de calcular una comisión.
3	Todo el sistema debe de estar disponible vía internet	5	Es importante que todo el sistema se encuentre disponible en cualquier lugar.
<b>Prioridad - Alto Nivel</b>			
4	Los profesores deben de ingresar sus horarios disponibles, dentro del sistema	5	Los profesores deben de registrar sus horarios disponibles para cada día, como mínimo de una hora
5	Los profesores deben de ingresar sus clases futuras y pasadas.	5	Cada clase debe de ser como mínimo de una hora
6	Permitir a estudiantes, con crédito, puedan registrar sus clases.	7	Tener un registro de los pagos y asignarle un costo por hora para clases de español, permitir si puede registrar las clases.
7	Registro y cálculo de costo de las traducciones.	1	Tener un registro de las traducciones dentro del instituto.
8	Correo electrónico por cada clase registrada a profesores, estudiantes y administración	2	Cuando se registre una clase enviar correo a: Administración, profesores y estudiantes
9	Sistema debe de tener soporte multilingüe.	3	Tener una interfaz de usuario tanto en español como en ingles
<b>Prioridad - Medio Nivel</b>			
10	No deben de existir choques de horario en el clases registradas, en los profesores	2	Las clases registradas no deben de haber choques
11	Los pagos a profesores solamente se realizaran si los profesores rellenan un reporte después de pasadas las clases.	1	Las clases deben de tener un informe con tres campos para llenar, este debe de llenarse despues de terminada la clase
12	Configuración del nuevo servidor web.	7	

13	Migración del web host, al nuevo	3	Migrar del anterior host al nuevo.
14	Cada clase debe de tener la posibilidad de añadir más de un estudiante	1	Las clases son de tipo individual como en grupos
15	Cada profesor debe de tener una diferente comisión por las clases pasadas	1	De acuerdo al tipo de contrato firmado tiene distinta comisión
16	Mostrar los horarios de clases por semana	3	Mostrar los horarios disponibles como de clases por semana
17	Registrar clientes de traducciones	½	Tener un registro de todos los clientes `por traducciones
18	Cada profesor podrá ver los horarios con otros profesores y poder ver lo que avanza con este profesor	1	Compartir la información que introducen entre distintos profesores
<b>Prioridad - Bajo Nivel</b>			
19	Tener un registro de las diferentes aulas del instituto	½	- 0 -
20	Tener un registro de los diferentes cursos que se ofrecen dentro del instituto	½	- 0 -
21	El sistema debe de diferenciar si el profesor está contratado o no.	½	- 0 -
22	El sistema debe de tener un registro de monedas y sus respectivos cambios de moneda.	2	Los pagos deben de realizarse bajo diferentes monedas.
<b>Esfuerzo total días:</b>		<b>~67</b>	

## ANEXO D – FORMULARIO DE REGISTRO DE ESTUDIANTES

Datos Personales / Personal Details	
Nombre Name	
Apellido Lastname	
Nacionalidad Nationality	
Fecha de Nacimiento (dd,mm,aaaa) Date of Birth	
Edad Age	
Sexo (masculino / femenino) Gender	
Correo – E e-mail	
Skype Skype Adress	
Domicilio Home Address	
Ciudad Town / City	
Provincia – Estado State / Region	
País Country	
Ocupación Occupation	
Número Telefónico telephone number	
¿Cuánto tiempo quiere estudiar español? (estimación) How long do you want to study Spanish? (estimate)	
¿Cuántos días por semana? How many days per week?	
¿Cuántas horas por día? How many hours per day?	
¿Qué nivel de español? (estimación) básico, intermedio, avanzado Which is your Spanish level? (estimate) elemental , intermediate, advanced	
¿Dónde estudió español? Where did you study Spanish?	
¿En qué aspectos desea concentrarse? lectura, gramática, conversación... On what areas do you want to focus?	
¿Cómo nos encontraste? How did you find us?	
Observaciones Suggestions	

## ANEXO E - PROGRAMA CONTENIDOS EN A1, NIVEL BASICO: 30 HORAS.

### - DATOS DEL CURSO

**Nombre del curso:** Curso de español presencial

**Código del curso:** esp-pr

**Descripción:** Curso de español presenciales, programa de 30 horas.

### - MÉTODO DE ENSEÑANZA

**Clases por Skype:** Si

**Clases presenciales:** No

**Actividades extracurriculares:** Si

**Otro tipo de enseñanza:** ninguno

### - CONTENIDO DEL CURSO

Unidad	Cont. Funcionales	Cont. Gramat.	Vocab.	Horas	Material
1	Saludos. Pronunciación.	Pronombres	Expresiones formales e informales. El alfabeto.	2	Planet 1 ejercicios Spanish for Mastery.
1	Presentaciones	Pronombres Interrogativos. Adj. posesivos	¿Quién soy yo?. Etc. Nacionalidades	2	IDEM
1	Objetos y lugares	Artículos determinados e indeterminados.	Sitios en una ciudad. Diferentes objetos comunes.	4	IDEM
1	Instrucciones	Imperativos afirmativos y negativos	Diferentes ordenes y sugerencias	2	Planet 2 ejercicios
1	Descripción Física	Adj. Descriptivos. Verbos SER y ESTAR	Valoraciones y Calificaciones.	2	IDEM 1, 2 y 3
2	Comparaciones	Verbos regulares en –AR Verbos regulares –er, -ir. Más/menos que + adj.	Partes del cuerpo. Actividades diarias. Verbos regulares y con cambio de radical	6	IDEM
2	Descripción de estado físico, carácter y	Verbos TENER y LLEVAR.	Otras actividades diarias	2	IDEM

	vestimenta de las personas	Expresión ACABAR DE + infinitivo			
2	Llamar la atención. Pedir disculpas.	Verbos reflexivos. El uso del impersonal SE	Escenas de la vida diaria. presente	4	Planet 1 y 2 ejercicios y Spanish for mastery
2	Verbos que expresan movimiento.	Participio presente	Actividades que se realizan en un determinado tiempo progresivo.	2	IDEM
3	Locaciones	Preposiciones de lugar compuestas. HAY, ESTA, ESTAN.	Lugares en la oficina., en la casa y en la clase.	2	Planet 1 y Planet ejercicios.
3	La Hora	Números. Expresiones de tiempo	La hora, Partes del día, Días de la semana, meses, años y fechas	2	IDEM anterior
				<b>30 horas</b>	

**Bibliografía utilizada:** es español 1, Planeta 1.

**Total de Horas:** 30 horas

**- MÉTODO DE EVALUACIÓN**

**Evaluación:** un examen semanal, al cumplir las 5 horas.

## CONTENIDO

<b>1. MARCO PRELIMINAR</b> .....	<b>4</b>
1.1 Introducción .....	4
1.2 Antecedentes .....	5
1.2.1 De la institución .....	5
1.2.2 Del proyecto.....	7
1.3 Análisis y planteamiento del problema .....	8
1.3.1 Análisis de la problemática .....	8
1.3.2 Planteamiento del problema .....	10
1.4 Objetivos .....	10
1.4.1 Objetivos Generales.....	10
1.4.2 Objetivos específicos .....	10
1.5 Justificaciones .....	11
1.5.1 Justificación Técnica-científica .....	11
1.5.2 Justificación Económica .....	12
1.5.3 Justificación Social .....	12
1.6 Métodos y técnicas .....	13
1.7 Alcances y aportes.....	13
1.7.1 Alcances .....	13
1.7.2 Aportes.....	14
<b>2. MARCO TEÓRICO</b> .....	<b>15</b>
2.1 El proceso unificado .....	15
2.1.1 Principios.....	15
2.1.2 Características.....	16
2.1.2.1 Proceso dirigido por Casos de Uso .....	16
2.1.2.2 Proceso centrado en la Arquitectura.....	17
2.1.2.3 Proceso iterativo e incremental .....	17
2.1.3 Estructura del proceso.....	18
2.1.4 Estructura dinámica del proceso, faces e iteraciones .....	19
2.1.5 Ciclos y fases .....	20
2.1.5.1 Inicio.....	21
2.1.5.2 Elaboración .....	21
2.1.5.3 Construcción .....	22
2.1.5.4 Transición.....	22
2.2 WEB 2.0 .....	23
2.2.1 Siete principios constitutivos de las aplicaciones Web 2.0 .....	23
2.3 Modelo Vista Controlador.....	24

2.3.1 Flujo de acciones dentro del MVC.....	25
2.4 Representational State Transfer (REST).....	26
2.5 AJAX.....	27
2.5.1 Navegadores web que permiten AJAX .....	29
2.6 Ruby on Rails .....	29
2.6.3 Arquitectura MVC de Rails.....	29
2.6.4 Módulos del framework Ruby on Rails.....	30
2.6.5 Soporte de servidores Web .....	31
2.6.6 Soporte de Bases de Datos .....	31
2.7 Subversion .....	31
2.7 Modelo de trabajo en Subversion copiar-modificar-mezclar.....	32
2.8 Modelo de proceso de la metodología Rup incorporando técnicas de desarrollo de software profesional y gestión del proyecto SCRUM. ....	33
2.8.1 Introducción.....	33
2.8.2 Fase de inicio .....	33
2.8.2.1 Requisitos.....	33
2.8.2.2 Casos de uso .....	34
2.8.2.2.1 Notación.....	35
2.8.3 Fase de Elaboración .....	36
2.8.3.1 Análisis .....	36
2.8.4 Fase de implementación.....	37
2.8.6 Calidad del software .....	42
2.8.7 Seguridad del software .....	46
2.8.7.1 Método de puntuación Dread .....	46
<b>3. MARCO APLICATIVO .....</b>	<b>48</b>
3.1 Fase de inicio.....	48
3.1.1 Análisis de requisitos .....	48
3.2.2 CASOS DE USO .....	49
3.3 Fase de elaboración .....	59
3.3 Análisis .....	59
3.3.1 Diagrama de paquetes.....	59
3.3.2 Diagrama de secuencias .....	60
3.4 fase de implementación .....	62
3.4.1 Diagrama de clases – Sistema Web integrado de planificación y pagos.....	62
3.4.2 Prototipado de navegación.....	63
3.4.3 Modelo de recursos del sistema.....	63
3.4.4 TÉCNICAS DE DESARROLLO DE SOFTWARE PROFESIONAL .....	65
3.4.4.1 Uso de frameworks de aplicaciones .....	65

3.4.4.1.1 Patrón de arquitectura MVC .....	65
3.4.4.1.2 AJAX .....	66
3.4.4.1.3 Manejo de versionado de código fuente .....	67
3.4.5 Gestión del proyecto .....	68
3.4.5.1 Definición de roles del SCRUM .....	68
3.4.5.2 Desarrollo de los sprints .....	68
3.4.5.2.2 Segundo sprint.....	69
3.4.5.2.3 Tercer sprint.....	70
3.4.5.2.4 Desarrollo del portal en cada iteración o sprint.....	71
<b>4. CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>78</b>
4.1 Conclusiones .....	78
4.2 Recomendaciones.....	79
<b>REFERENCIA BIBLIOGRÁFICA:.....</b>	<b>80</b>
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>82</b>
<b>ANEXO A – SCRUM: MODELO DE DESARROLLO ÁGIL DE PRODUCTOS.....</b>	<b>83</b>
A.1 Roles y actores.....	83
A.2 Definición de Roles:.....	84
A.3 Flujo de trabajo.....	85
<b>ANEXO B - AJAX Y LA USABILIDAD.....</b>	<b>87</b>
B.1 Problemas .....	88
<b>ANEXO C - PRODUCT BACKLOG FINAL .....</b>	<b>90</b>
<b>ANEXO D – FORMULARIO DE REGISTRO DE ESTUDIANTES .....</b>	<b>92</b>
<b>ANEXO E - PROGRAMA CONTENIDOS EN A1, NIVEL BASICO: 30 HORAS. ....</b>	<b>93</b>
<b>LISTADO DE FIGURAS.....</b>	<b>98</b>
<b>LISTADO DE TABLAS.....</b>	<b>99</b>

## LISTADO DE FIGURAS

Figura 1.1 Organigrama del Instituto Exclusivo de Idiomas .....	7
Figura 2.1 Mini Cascada dentro de cada iteracion .....	18
Figura 2.2 Fases del desarrollo RUP .....	19
Figura 2.3 Fases e hitos en RUP .....	20
Figura 2.4 Distribucion tipica de recursos humanos .....	20
Figura 2.5 Arquitectura del patron modelo-vista-controlador .....	25
figura 2.6 Modelo tradional y Ajax .....	28
Figura 2.7 Modelo de ejecucion de la plataforma ruby on rails.....	30
figura 2.8 Representacion grafica de paquetes.....	37
Figura 2.9 Representación de una clase.....	38
Figura 2.10 Generalización en UML.....	39
Figura 2.11 Asociacion de clases .....	40
Figura 3.1 Diagrama de casos de uso principal .....	49
Figura 3.2 Diagrama de casos de uso Registro de actividades pedagógicas.....	50
Figura 3.3 Casos de uso registro y Generacion de pagos .....	51
Figura 3.4 Casos de uso manejo de traducciones y clientes de traducciones .....	51
figura 3.5 Diagrama de paquetes .....	60
figura 3.6 Diagrama de secuencia Registro de nueva actividad pedagogica .....	60
figura 3.7 Diagrama de secuencia Generacion de pagos a profesores .....	61
figura 3.8 Diagrama de secuencia Registro de traducciones y gestion de la traduccion.....	61
figura 3.9 DIAGRAMA DE CLASES DEL PROYECTO .....	62
figura 3.10 Funcionalidad dentro del primer sprint, mostrar horarios por semana (Profesor)71	
figura 3.11 Funcionalidad dentro del segundo sprint, Soporte de ajax.....	71
figura 3.12 Funcionalidad dentro del Tercer sprint, Rediseño del sitio .....	72
figura 3.13 Funcionalidad dentro del Tercer sprint, añadir una nueva clase mediante Ajax y soporte multilingüe (Interfaz profesor).....	72
Figura a.0.1 Ciclo de desarrollo dentro de Scrum .....	85
Figura a.0.1 Aplicación web que utiliza ajax (gmail).....	87

## LISTADO DE TABLAS

Tabla 2.1 Distribucion tipica de esfuerzo y tiempo .....	20
Tabla 2.2 Comparacion entre el diseño tradicional y REST .....	27
Tabla 2.3 Notacion de casos de uso .....	35
Tabla 2.4 Descripcion de actores.....	35
Tabla 2.5 Descripcion de casos de uso .....	36
Tabla 2.6 notacion de los diagramas de interaccion .....	36
Tabla 2.7 Descripcion de asignacion de recursos REST dentro de una aplicacion Web .....	40
Tabla 2.8 Modelo de repartición de tareas dentro de un sprint.....	42
Tabla 2.9 Factores y MEtricas.....	44
Tabla 2.10 Valoracion Metrica Punto Funcion .....	45
Tabla 2.11 Valoracion metrica punto funcion .....	45
Tabla 2.12 amenazas comunes dentro de aplicaciones web .....	46
Tabla 2.13 Descripcion del mecanismo DREAD.....	47
Tabla 2.14 Descripcion de las ponderaciones para aplicar DREAD .....	47
Tabla 3.1 Requisitos candidatos .....	49
Tabla 3.2 Descripcion del actor estudiante .....	52
Tabla 3.3 Descripcion del actor Profesor.....	52
Tabla 3.4 Descripcion del actor Secretaria .....	52
Tabla 3.5 Descripcion del actor traductor .....	52
Tabla 3.6 Caso de uso AUTENTIFICarse.....	53
Tabla 3.7 Caso de uso REGISTRARSE dentro del SISTEMA .....	53
Tabla 3.8 Caso de uso Activar cuenta de usuario.....	53
Tabla 3.9 ELABORAR PROGRAMA Y CONTENIDO DEL CURSO.....	54
Tabla 3.10 Caso de uso REGISTRA CLASES.....	54
Tabla 3.11 Caso de uso ACTUALIZA EL CREDITO DEL ESTUDIANTE.....	55
Tabla 3.12 Caso de uso ACTUALIZA EL CREDITO DEL ESTUDIANTE.....	55

Tabla 3.13 Llenar Y COMPARTIR EL PERFIL DE USUARIO .....	55
Tabla 3.14 ELABORAR Y COMPARTIR EL INFORME DE LA CLASE .....	56
Tabla 3.15 Caso de uso REGISTRA HORARIOS DISPONIBLES .....	56
Tabla 3.16 Caso de uso EXAMINA HORARIOS DE PROFESOR.....	56
Tabla 3.17 Caso de uso REGISTRAR Y CALCULAR COSTO DE TRADUCCIONES .....	57
Tabla 3.18 Caso de uso SEGUIMIENTO Y ENVÍO DE LA TRADUCCIÓN .....	57
Tabla 3.19 Caso de uso REGISTRAR Y GESTIONAR CLIENTES .....	58
Tabla 3.20 Caso de uso ASIGNA CREDITO A ESTUDIANTES.....	58
Tabla 3.21 Caso de uso REALIZA PAGOS A PROFESORES .....	58
Tabla 3.22 Caso de uso GENERAR INFORMES DE INGRESOS Y GASTOS .....	59
Tabla 3.23 Diseño navegacional del sistema.....	63
Tabla 3.24 Definición del modelo de recursos, REST .....	64
Tabla 3.25 Primer Sprint backlog.....	69
Tabla 3.26 Sprint Backlog del segundo sprint .....	70
Tabla 3.27 Sprint Backlog del Tercer sprint.....	70
Tabla 3.28 Ejemplo de una prueba unitaria en lenguaje de programacion ruby.....	73
Tabla 3.29 Resultado de metrica de Mc Call .....	74
Tabla 3.30 Calculo de puntos de características .....	75
Tabla 3.31 Calculo del factor de ajuste .....	76
Tabla 3.32 Calidad utilizando Punto Funcion .....	76
Tabla 3.33 Evaluacion de las vulnerabilidades segun DREAD .....	77
Tabla A.1 Historia de la gallina y el cerdo.....	83