



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

**SISTEMA DE ADMINISTRACIÓN DE PROYECTOS PARA LA
EMPRESA VERBUX SYSTEMS**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN
COMPUTACIÓN

EMILIO ANDRÉS PONCE RÍOS

PROFESOR GUÍA
JOSÉ ALBERTO PINO URTUBIA

MIEMBROS DE LA COMISIÓN
MARIA CECILIA BASTARRICA PIÑEYRO
DIONISIO ALEJANDRO GONZALEZ GONZALEZ

SANTIAGO
OCTUBRE 2009

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN
POR: EMILIO PONCE RÍOS
PROF. GUIA: Sr. JOSE A. PINO URTUBIA

“SISTEMA DE ADMINISTRACIÓN DE PROYECTOS PARA LA
EMPRESA VERBUX SYSTEMS”

El objetivo principal del presente trabajo fue desarrollar un sistema de administración de proyectos para la empresa Verbux.

Antes del resultado de esta memoria, Verbux no contaba con un seguimiento estándar para los proyectos, y tampoco con un proceso uniforme para su desarrollo. De hecho, es difícil buscar información sobre los proyectos pasados o de cuánto se ha invertido en uno en curso.

Se analizaron las distintas opciones posibles de solución, tomando en cuenta los requerimientos de los usuarios y las restricciones de la empresa, para decidir cuál sería la óptima para su implementación. Luego del análisis y comparación de las opciones se determinó la mejor entre todas ellas, considerando la instancia particular de empresa pequeña y las características que tiene Verbux.

Se presenta el desarrollo de un sistema web ad-hoc para la empresa, que privilegia la extensibilidad, modularidad y facilidad en la programación por sobre otras características. El framework de desarrollo es CakePHP, y el motor de base de datos es PostgreSQL herramientas que sustentan una plataforma de programación sencilla, pero robusta.

Concluyendo, se desarrolló un sistema basado en las necesidades y requerimientos de la empresa sobre una plataforma práctica y confiable, usando solamente software de carácter libre para su programación e implementación. Este desarrollo le entrega a la empresa una herramienta que le permitirá disminuir tiempos de administración, gestión y consulta de los proyectos en curso, junto con poder tener referencias rápidas y claras de los proyectos ya cerrados.

Agradecimientos

Es difícil agradecer a todas las personas que por ocho años me impulsaron a seguir adelante con este hermoso sueño que finalmente se concretó.

Agradezco a toda mi familia, en especial a mis padres y hermanas que gracias a su esfuerzo y paciencia me brindaron la tranquilidad para completar mi carrera y esta memoria.

También quiero agradecer a Mirla mi amada compañera, la cual conozco de casi toda la vida, pero que desde aquel especial verano decidimos compartirla juntos, agradezco tu ternura, comprensión y apoyo que me diste cuando los necesité.

Agradezco todo el apoyo brindado por mi profesor guía José Alberto Pino U. quien supo exigir y moderar los esfuerzos según se debía. También debo agradecer a la empresa Verbux, representada por Marco Rocco G., por el apoyo y disposición prestados en la realización de este proyecto.

Finalmente agradecer a todos los que de una u otra manera interfirieron durante todo este tiempo para que este objetivo se cumpliera.

Gracias a todos, Emilio Ponce R.

Índice

1. Introducción	6
1.1. Justificación	6
1.2. Objetivo General	8
1.3. Objetivos Específicos	8
2. Revisión Bibliográfica	9
2.1. ERP	9
2.1.1. SAP ERP	10
2.1.2. Openbravo ERP	12
2.2. PMS	13
2.2.1. Tipos de softwares para la gestión de proyectos	14
2.3. Desarrollo de un sistema a medida	16
2.3.1. Lenguaje	16
2.3.2. Base de datos	19
2.3.3. Framework	22
3. Plan de Trabajo	27
3.1. Metodología	27
3.2. Diseño	27
3.3. Construcción	27
3.4. Validación	28
3.5. Integración	28
3.6. Validación del sistema	28
4. Diseño del sistema	29
4.1. Selección de personas a entrevistar	29
4.2. Especificación de requerimientos	29
4.2.1. Requisitos Funcionales	30
4.2.2. Requisitos No Funcionales	31
4.3. Arquitectura Web	32
4.4. Cakephp y MVC	32
4.5. Arquitectura de Datos	33
4.5.1. Personas	34
4.5.2. Proyectos	35
4.5.3. Distribución y Aprobación	37
4.5.4. Ingresos y Egresos	38

4.5.5. Modelo Completo	39
5. Construcción del sistema	41
5.1. Roles	42
5.2. Modelos	42
5.3. Vistas	45
5.4. Controladores	46
6. Descripción del sistema	48
6.1. Interfaz de marcado de horas (Reloj)	48
6.2. Interfaz del sistema	49
6.2.1. Distribución de horas	49
6.2.2. Aprobación de horas	50
6.2.3. Generación de bitácora	51
6.2.4. Informes	52
6.2.5. Administración del sistema	53
6.2.6. Creación y administracion de proyectos	54
6.3. Resultados	55
7. Conclusiones	57
7.1. Contribuciones	57
7.2. Trabajo Futuro	59
A. Requisitos	63
A.1. Requisitos de usuario	63
A.1.1. Requisitos funcionales	63
A.1.2. Requisitos de calidad	68
A.1.3. Requisitos de Restricción	69
A.2. Requisitos de software	71
A.2.1. Requisitos funcionales	71
B. Matriz de trazabilidad, requisitos de usuarios	75
C. Matriz de trazabilidad, requisitos de software	76
D. Glosario	77
E. Comparativa de frameworks	79

Índice de figuras

1.	Componentes de PostgreSQL	20
2.	Rendimiento de frameworks	23
3.	Modelo MVC	25
4.	Arquitectura Web Tres Capas	32
5.	Grupo de entidades Personas	34
6.	Grupo de entidades Proyectos	35
7.	Grupo de entidades de distribución y aprobación	37
8.	Grupo de entidades Ingresos y Egresos	39
9.	Modelo completo de entidades	40
10.	Diagrama de clases importantes	43
11.	Layout del sistema	45
12.	Interfaz de marcado de horas	48
13.	Distribución de horas	49
14.	Aprobación de horas	50
15.	Bitácora	51
16.	Informes libres	53
17.	Administración de personas	54
18.	Agregar ingresos	55
19.	Módulo Proyectos	56
20.	Comparativa de frameworks PHP	80

Índice de tablas

1.	Límites de PostgreSQL	21
2.	Rendimiento de frameworks	23
3.	Tipo de relaciones en CakePHP	44
4.	Tiempos ahorrados	58
5.	Matriz de trazabilidad, requisitos de usuario	75
6.	Matriz de trazabilidad, requisitos de software	76

1. Introducción

Verbux Systems Ltda. [1] (en adelante sólo ‘Verbux’), es una empresa comercial de aproximadamente 15 empleados. Sus áreas de actividad son: outsourcing de computación para empresas, mantenimiento y armado de redes y computadores, y finalmente desarrollo de software.

Verbux se especializa tanto en el mantenimiento de redes como en el soporte de la mismas al usuario, por lo que generalmente un mismo cliente puede ser parte de varios proyectos, y en general, proyectos de distinto tipo.

Debido al aumento de proyectos y clientes (con clientes de otras regiones), Verbux tiene como desafío el poder controlar el estado de avance de los proyectos. Naturalmente, para hacerlo, Verbux debe contar con un sistema interconectado entre sus diversas oficinas.

En consecuencia, el principal motivo de esta memoria es plantear una solución a los distintos tipos de problemas que se tiene actualmente con el manejo de los proyectos.

1.1. Justificación

Un punto importante en la gestión de los proyectos es el manejo de las necesidades de éstos (materiales, recursos, etc.), como en las obligaciones, requerimientos o compromisos adquiridos con el cliente.

Actualmente, los jefes de proyecto no pueden saber el estado de avance de los proyectos a su cargo, a menos que los inspeccione *in situ* o se le esté preguntando constantemente al encargado de la realización, generando así demoras y tiempos muertos en el desarrollo.

La visibilidad de los proyectos para los gerentes es prácticamente nula, por ejemplo, para poder saber cómo se están desarrollando o los problemas que han surgido, es necesario reunirse con los jefes de proyecto y preguntarles directamente. Esto también genera demoras en la información, impidiendo así la posibilidad de tomar medidas correctivas oportunas para mejorar la gestión.

También, con el sistema actual es posible sumar todos los ingresos y restar gastos de la empresa para saber la situación financiera final, pero el resultado no aporta ningún tipo de información adicional, sólo si el mes fue rentable.

En la actualidad, el seguimiento de los proyectos se hace usando planillas de Microsoft Excel con los datos de ingresos y gastos, las horas asociadas y los requerimientos de cada proyecto.

La información proviene de una gama muy amplia de fuentes, por lo que para poder tener una fotografía de la administración del proyecto (gastos, requerimientos, nivel de avance, etc.), hay que hacer un catastro de todos los documentos relacionados, lo cual suele ser mucho trabajo.

Realizar un balance de los proyectos en cualquier momento toma demasiado tiempo, y el de la empresa completa aún más, por lo que rara vez se efectúa, pero sin ese balance es muy difícil conocer si los datos proyectados concuerdan con los reales.

Un mayor detalle de esta información sería de mucha utilidad para la empresa, ya que se podría tener estadísticas de los progresos por proyecto y general de la empresa. Idealmente se querría saber qué proyectos son los más rentables, e incluso, si todos los proyectos lo son. Además, proporcionaría la oportunidad para realizar cambios correctivos.

Existen softwares licenciados y opensource que pueden satisfacer los requerimientos antes descritos de manera robusta y rápida, pero hay varias limitantes por las cuales los ejecutivos de Verbux no optaron por software de estos tipos.

Los softwares licenciados que cumplen estos requisitos son demasiado costosos, y en general, su uso e implementación alcanza su nivel de productividad más alto con una cantidad de usuarios media a grande, que no es el caso de Verbux.

Los softwares opensource son difíciles de adecuar a las necesidades de la empresa y, en general, la empresa tiene que adecuarse a las posibilidades del software.

Junto con tener la experiencia de haber trabajado softwares licenciados a través de las empresas a las cuales Verbux presta servicio, y referencias sobre los sistemas opensource, la gerencia desestimó su implementación.

Los sistemas PMS (Project Management Software) son estables y seguros pero no cumplen con todos los requerimientos que la empresa ha puesto como indispensables, algunos de ellos son:

1. Posibilidad de tener distintos roles dentro del mismo sistema.
2. Obtener resultados del estilo, quién trabajo en qué o cuántas horas libres tiene un empleado mensualmente.
3. Adaptar el sistema al funcionamiento de la empresa.

Por último, se decidió desarrollar un sistema de administración de proyectos, para así tener una herramienta que cumpla las necesidades técnicas y presupuestarias de la empresa, de una manera precisa y práctica.

1.2. Objetivo General

El objetivo general de esta memoria es diseñar e implementar un sistema web que permita a la empresa Verbux, llevar un control más acabado de sus proyectos, para poder así tomar decisiones correctivas si fuese el caso.

1.3. Objetivos Específicos

Algunos objetivos específicos que se derivan del objetivo general son:

1. Modelar el servicio que entrega la empresa Verbux a sus clientes, junto con estandarizar las distintas etapas por las que pasa un proyecto.
2. Implementar un sistema automatizado de reportabilidad útil para tomar decisiones en la empresa y aplicar medidas correctivas.
3. Coordinar y distribuir mejor los tiempos y recursos de la empresa a través de la aplicación web.
4. Controlar los gastos e ingresos de la empresa, así como también los distintos procesos que se llevan a cabo diaria y mensualmente en Verbux.

2. Revisión Bibliográfica

En este capítulo veremos qué tipo de sistema se desarrolló y analizaremos dos aplicaciones de control de gestión de proyectos que pudieron satisfacer nuestros requerimientos, uno licenciado y otro opensource.

Otra posible opción de solución fue la implementación de un sistema PMS, veremos sus ventajas y desventajas, y el por qué no se seleccionó este tipo de sistemas.

Se evaluó el desarrollo de un sistema a medida, junto con el lenguaje que se usó para construir la aplicación, la base de datos en la cual se implementó el diseño, además de elegir un framework que adoptara este lenguaje y soportara nuestra base de datos.

Finalmente resolvimos qué solución se implementara en Verbux dadas las condiciones de la empresa junto con las ventajas y restricciones de cada opción.

2.1. ERP

Los sistemas de planificación de recursos de la empresa (en inglés ERP [5], Enterprise Resource Planning) son sistemas de gestión de información, que integran y automatizan muchas de las prácticas de negocio asociadas con los aspectos operativos o productivos de una empresa.

La adopción de un ERP es un proceso complejo, conformado por etapas bien diferenciadas:

1. Establecer la necesidad del cambio: se tiene que tomar en cuenta que el plan de cambio incluye el reemplazo de todas las herramientas que soportan los procesos de negocios. Se debe verificar la situación actual y complementarla con los objetivos del negocio.
2. Selección de la solución: para esta etapa es necesario definir los responsables de las tomas de decisiones, los criterios de evaluación de las posibles soluciones, el análisis de las alternativas, así como evaluar y elegir la que mejores beneficios brinde a la organización.
3. Implementación de la Solución.
4. Puesta en Marcha.
5. Impacto en la Organización.

Hoy en día, existe una gran variedad de sistemas que cumplen esta definición y su uso se ha ido expandiendo considerablemente, principalmente por su estabilidad y confiabilidad.

2.1.1. SAP ERP

SAP AG (Systeme, Anwendungen und Produkte; Sistemas, Aplicaciones y Productos), es el más importante proveedor de software empresarial en el mundo. Como empresa, comercializa un conjunto de aplicaciones de software para soluciones integradas de negocios, entre ellas su ERP 'SAP ERP' [6].

SAP comercializa sus productos en diferentes industrias y países del mundo, estas van desde compañías privadas hasta gubernamentales. Algunas de las industrias en las que SAP esta presente son:

1. Materias primas, minería y agricultura.
2. Gas y petróleo.
3. Químicas.
4. Metalúrgicas.
5. Farmacéuticas.
6. Materiales de construcción, arcilla y vidrio.
7. Construcción pesada.
8. Servicios.
9. Consultorías y software.
10. Sanatorios y hospitales.
11. Muebles.
12. Automoción.
13. Textil y vestidos.
14. Papel y maderas.
15. Sector Público.

16. Retail.
17. Educación Superior e Investigación.
18. Informática.
19. Medios.

SAP ERP, por su diseño modular, ofrece indistintamente la habilitación de módulos para la gestión dentro de la empresa. La idea de este diseño es hacer un sistema escalable y confiable.

- Gestión Financiera (FI): libro mayor, libros auxiliares, ledgers especiales, etc.
- Controlling (CO): gastos generales, costos de producto, cuenta de resultados, centros de beneficio, etc.
- Tesorería (TR): control de fondos, gestión presupuestaria, flujo de efectivo.
- Sistema de proyectos (PS): grafos, contabilidad de costos de proyecto, etc.
- Gestión de personal (HR): gestión de personal, cálculo de la nómina, contratación de personal, etc.
- Mantenimiento (PM): planificación de tareas, planificación de mantenimiento, etc.
- Gestión de calidad (QM): planificación de calidad, inspección de calidad, certificado de aviso de calidad, etc.
- Planificación de producto (PP): fabricación sobre pedido, fabricación en serie, Kanban, etc.
- Gestión de material (MM): gestión de stocks, compras, verificación de facturas, etc.
- Comercial(SD): ventas, expedición, facturación, etc.
- Workflow (WF), soluciones sectoriales (IS): contienen funciones que se pueden aplicar en todos los módulos.
- Activos Fijo(AF), ingresos, depreciación y amortización de activos fijos. (Sub módulo de FI) AM.

SAP ERP es un sistema muy estable, seguro y es uno de los ERP más usados en Chile y el mundo, sin embargo, la gerencia de la empresa desestimó su uso por:

- El alto valor de las licencias.
- La dificultad de adaptación debido a la rigidez del sistema.
- El bajo nivel de utilidad funcional.

El costo de las licencias lo podemos apreciar en que sólo por la habilitación de cualquier módulo de este sistema, el costo asciende a varios miles, cientos de miles, y a veces, millones de dólares.

La dificultad de adaptación se debe a que los sistemas rígidos de monitoreo de proyectos implementados en SAP no abarcan toda la gama de proyectos existentes en Verbux (por la experiencia adquirida desde los clientes de Verbux).

El bajo nivel de utilidad funcional, particularmente para Verbux, está dado porque algunos de los módulos implementados en SAP, el comercial por ejemplo, abarcan más funciones de las que Verbux adoptaría.

2.1.2. Openbravo ERP

OpenBravo ERP [9] es una aplicación de código abierto de gestión empresarial, destinada a empresas de pequeño y mediano tamaño, rango en que se haya Verbux.

Las grandes áreas que integra actualmente el sistema de gestión son:

- Gestión de los datos maestros: productos, componentes, listas de materiales, clientes, proveedores, etc.
- Gestión de los aprovisionamientos: tarifas, pedidos de compra, recepción de mercancías, verificación de facturas de proveedores, evaluación de proveedores, etc.
- Gestión de almacenes: almacenes y ubicaciones, unidades de almacén, lotes, número de serie, bultos, etiquetas, entradas, salidas, movimientos entre almacenes, inventarios, valoración de existencias, transportes, etc.
- Gestión de proyectos: proyectos, fases, presupuestos, gastos, compras asociadas, etc.

- Gestión de servicios: recursos, servicios, gastos, gastos refacturables, facturación de servicios, nivel de servicio, etc.
- Gestión de la producción: estructura de planta, hojas de ruta y BOM's, órdenes de fabricación, partes de trabajo, incidencias de trabajo, partes de mantenimiento, etc.
- Gestión comercial y gestión de las relaciones con clientes (CRM): pedidos de venta, tarifas, albaranes, facturación, comisiones, CRM, etc.
- Gestión económico-financiera: plan de cuentas, cuentas contables, impuestos, contabilidad general, cuentas a pagar, cuentas a cobrar, contabilidad bancaria, balance, cuenta de resultados, activos fijos, etc.
- Business Intelligence (BI): reporting, análisis multidimensional (OLAP), cuadros de mando predefinidos.

Este sistema, tiene bastante prestigio frente a sus similares opensource, pero, al igual que el sistema anterior, la gerencia de la empresa desestimó su uso, debido a que la rigidez de OpenBravo fuerza a la empresa a adaptarse a él y no viceversa.

2.2. PMS

Software de administración de proyectos (en inglés Project management software PMS [7]) es un concepto que describe varios tipos de software, incluyendo softwares de programación, asignación de recursos, colaboración, comunicación y sistemas de documentación, utilizados para ayudar a organizar un proyecto complejo en diferentes tareas y en un tiempo determinado.

Planificación Una de las tareas más comunes, es programar o agendar una serie de eventos, y la complejidad de esta tarea puede variar considerablemente dependiendo de cómo se utiliza la herramienta. Algunos retos comunes incluyen:

- Coordinar, eventos que dependen unos de otros de diferentes formas.
- Programación de las personas a trabajar, y los recursos requeridos por las diversas tareas, comúnmente denominada programación de recursos.
- Hacer frente a las incertidumbres en las estimaciones de la duración de cada tarea.
- La organización de tareas para cumplir con diversos plazos.

- Coordinar múltiples proyectos simultáneamente a fin de satisfacer una variedad de requisitos.

Cálculo de la ruta crítica En muchos proyectos complejos, habrá una trayectoria crítica, o series de acontecimientos que dependan uno del otro, y que sus duraciones determinen directamente la longitud del proyecto entero. Algunos usos del software (por ejemplo, soluciones de la matriz de la estructura de la dependencia) pueden destacar estas tareas que son a menudo un buen candidato a cualquier esfuerzo de la optimización.

El proveer información La planificación de proyectos de software debe proporcionar una gran cantidad de información a distintas personas, justificando así el tiempo que lleva usarla. Requisitos típicos podrían incluir:

- Listas de tareas para las personas, los horarios y la asignación de recursos.
- Información general sobre cuánto tiempo llevará completar las tareas.
- Alerta temprana de los riesgos para el proyecto.
- Información sobre la carga de trabajo.
- Información histórica sobre cómo los proyectos han avanzado y, en particular, cómo la forma real y la proyectada de ejecución están relacionadas.
- La utilización óptima de los recursos disponibles.

2.2.1. Tipos de softwares para la gestión de proyectos

De escritorio La gestión de proyectos de software se puede implementar como un programa que se ejecuta en el escritorio de cada usuario.

Las aplicaciones de escritorio suelen almacenar sus datos en un archivo, aunque algunos tienen la capacidad de colaborar con otros usuarios, o para almacenar sus datos en una base de datos central. Incluso las aplicaciones basadas en archivos pueden compartir el plan del proyecto entre los usuarios siempre que esté en una unidad de red y sólo un usuario acceda a ella a la vez.

Aunque las aplicaciones de escritorio pueden ser escritas para que se ejecuten en ambientes heterogéneos, de múltiples sistemas operativos, esto es inusual.

Aplicaciones Web Un software de gestión de proyectos puede ser implementado como una aplicación Web, accesible a través de una intranet o extranet utilizando un navegador web. Esto tiene todas las ventajas y los inconvenientes habituales de las aplicaciones web, entre ellos:

- Se puede acceder desde cualquier tipo de computador sin necesidad de instalar un software.
- Facilidad de control de acceso.
- Multi-usuario.
- Sólo una versión de software e instalación que mantener.
- Típicamente más lento para responder que las aplicaciones de escritorio.
- La información del proyecto no está disponible cuando el usuario (o servidor) está offline.

Personal Las aplicaciones de gestión de proyecto personales son utilizadas en el hogar, por lo general, para el estilo de vida o para la gestión de proyectos en el hogar. Existe un considerable traslape con los sistemas de usuario único, aunque los software de gestión de proyectos personales generalmente tienen interfaces más sencillas.

Usuario Único Un sistema de usuario único está programado suponiendo que sólo una persona necesita modificar el plan del proyecto a la vez. Esto puede ser usado en pequeñas empresas, o en las que sólo unas pocas personas están involucradas en la planificación de proyectos. Las aplicaciones de escritorio en general, entran en esta categoría.

Colaborativo Un sistema colaborativo está diseñado para soportar a varios usuarios a la vez, modificando diferentes secciones del plan. Generalmente son basados en herramientas Web, incluyendo servidores externos, pero tiene la limitación de que sólo pueden utilizarse cuando el usuario tiene acceso a Internet.

Para hacer frente a esta limitación, existen herramientas de software cliente-servidor basadas en un cliente que se ejecuta en los clientes, computadores de escritorio, replicando la información de proyectos y tareas a otros miembros del equipo a través de un servidor central.

Algunas herramientas permiten a los miembros del equipo guardar sus planificaciones, para trabajar sobre ellas aunque no estén conectados a la red. Al conectarse a la base de datos, los cambios se sincronizarán con las otras planificaciones.

Integrado Un sistema integrado combina la gestión de proyectos o planificación de proyectos, con muchos otros aspectos de la vida de las empresas. Por ejemplo, pueden haber proyectos de seguimiento de fallos de problemas asignados a cada proyecto, la lista de clientes se convierte en un módulo de proyecto de gestión de relaciones con clientes, y cada persona en el proyecto tiene su propio plan de listas de tareas, calendarios y mensajes asociados con la funcionalidad de sus proyectos.

Del mismo modo, herramientas especializadas como SourceForge [8] integran la gestión de proyectos de software con control del código fuente (CVS) y el software de seguimiento de fallos, de modo que cada pieza de información se puede integrar en el mismo sistema.

Los sistemas PMS tienen gran aceptación en las empresas nacionales, ya que cumplen su función, el de monitorear el avance de los proyectos y estimar gastos y tiempos. Algunos de estos sistemas con bastante tiempo en el mercado, por lo que su estabilidad y confiabilidad es de un excelente nivel.

Esto cumple un muy buen porcentaje de los requerimientos, pero no los cumple todos, ya que aparte del control que se quiere llevar de los proyectos, se necesita información de cada empleado, a través de un registro de entradas y salidas, junto con esto se necesita saber cuáles fueron sus actividades y cuánto tiempo les tomó.

Con estos antecedentes podemos decir que los PMS no cumplen todos los requerimientos que se necesitan para Verbox, excluyendo esta posibilidad de la solución final.

2.3. Desarrollo de un sistema a medida

2.3.1. Lenguaje

La elección del lenguaje en el cual se desarrolló el sistema se hizo en base a la arquitectura que tiene la empresa Verbox ya montada y los requerimientos impuestos por el gerente a cargo.

Se evaluaron varios lenguajes de programación, JAVA, PHP, .NET entre otros, se analizó su integración con los sistema de la empresa y el impacto que generaría un sistema desarrollado en ese lenguaje.

Se escogió el lenguaje PHP [10] por la gran cantidad de herramientas que existen para su desarrollo [11], por la arquitectura física que tiene Verbox y porque es una herramienta libre para la construcción y desarrollo.

Se descartaron lenguajes como JAVA [12] y .NET [13] por su curva de aprendizaje que implicaba invertir una gran cantidad tiempo en el estudio del lenguaje y la costosa integración en la empresa de un lenguaje que actualmente Verbox no ocupa.

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Visión general PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida.

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, aunque el número de sitios en PHP ha compartido algo de su preponderante sitio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web. La versión más reciente de PHP es la 5.3.0 (para Windows) del 30 de junio de 2009.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Aunque todo en su diseño está orientado a facilitar la creación de página web, es posible crear aplicaciones con una interfaz gráfica para el usuario, utilizando la extensión PHP-Qt o PHP-GTK. También puede ser usado desde la línea de comandos, de la misma manera como Perl o Python pueden hacerlo, a esta versión de PHP se la llama PHP CLI (Command Line Interface).

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

PHP es una alternativa a las tecnologías de Microsoft ASP y ASP.NET (que utiliza C# VB.NET como lenguajes), a ColdFusion de la compañía Adobe (antes Macromedia), a JSP/Java de Sun Microsystems, y a CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un IDE (entorno de desarrollo integrado) comercial llamado Zend Studio. Recientemente, CodeGear (la división de lenguajes de programación de Borland) ha sacado al mercado un entorno integrado de desarrollo para PHP, denominado Delphi for PHP.

Existe un módulo para Eclipse [14], uno de los IDE más populares, que fué el que se ocupó para el desarrollo de la aplicación que da origen a esta memoria.

2.3.2. Base de datos

Se escogió implementar el modelo relacional de la base de datos en PostgreSQL [22] ya que versus su natural competidor como es MySQL [21] se comporta mejor con consultas anidadas, restricciones y múltiples procesadores [17] [18] [19] [20], arquitectura en la cual se implantó finalmente el sistema. En las comparativas revisadas, el rendimiento de PostgreSQL es mayor a MySQL para consultas complejas, por otra parte, la escalabilidad que da PostgreSQL al añadir nuevo hardware supera ampliamente a la de MySQL.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y de código abierto. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones compite a la par con bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

En la Figura 1 podemos ver la composición lógica del sistema administrador de base de datos PostgreSQL. Los componentes más importantes son:

- Aplicación cliente: esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP ó sockets locales.
- Demonio postmaster: este es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargarán de autenticar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- Ficheros de configuracion: los tres ficheros principales de configuración utilizados por PostgreSQL, postgresql.conf, pg_hba.conf y pg_ident.conf.
- Procesos hijos postgres: procesos hijos que se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes
- PostgreSQL share buffer cache: memoria compartida usada por PostgreSQL para almacenar datos en caché.

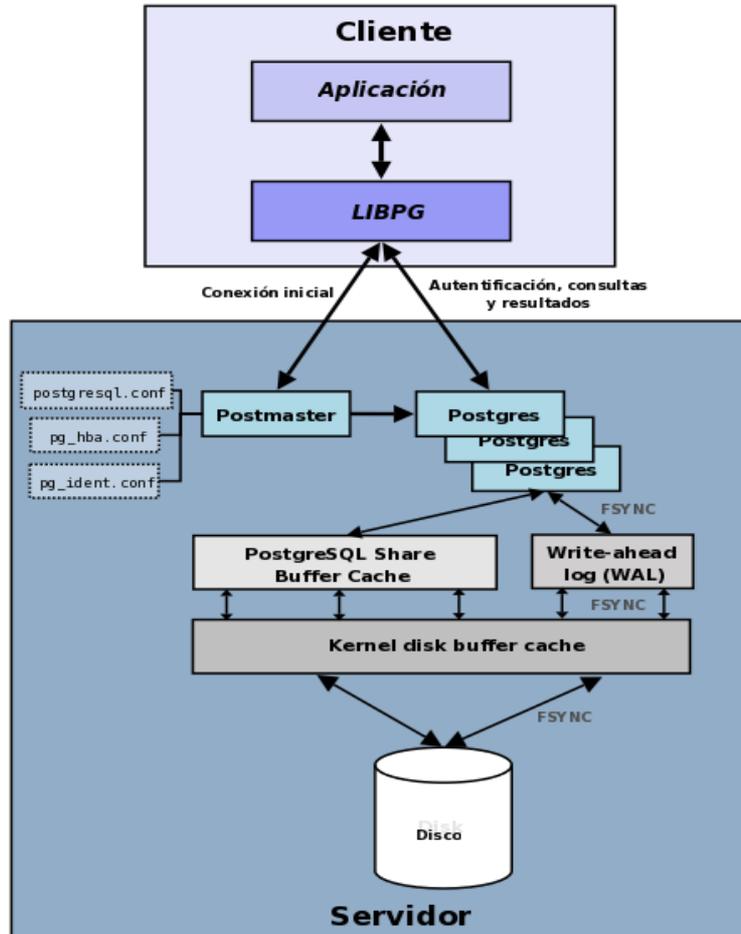


Figura 1: Componentes de PostgreSQL

- Write-Ahead Log (WAL): componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO).
- Kernel disk buffer cache: caché de disco del sistema operativo.
- Disco: disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione.

Características, Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo comenzó hace más de 15 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares, han sido las características que más se han tenido en cuenta durante su desarrollo.

Límite	Valor
Máximo tamaño base de dato	Ilimitado
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 TB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250 - 1600
Máximo número de índices por tabla	Ilimitado

Tabla 1: Límites de PostgreSQL

Algunas de las características más importantes y soportadas por PostgreSQL son:

Generales

- Es una base de datos 100 % ACID.
- Integridad referencial.
- Tablespaces.
- Replicación asíncrona.
- Copias de seguridad en caliente (Online/hot backups).
- Unicode.
- Juegos de caracteres internacionales.
- Multi-Version Concurrency Control (MVCC).
- Acceso encriptado vía SSL.
- Completa documentación.
- Licencia BSD.
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows.

Programación / Desarrollo

- Funciones o procedimientos almacenados (stored procedures) en numerosos lenguajes de programación, entre otros PL/pgSQL, PL/Perl, PL/Python y PL/Tcl.
- Numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos, tenemos disponibles, entre otros, tipos geométricos, de direcciones de red, de cadenas binarias, UUID, XML, matrices, etc.
- Soporta el almacenamiento de objetos binarios grandes: gráficos, videos, sonido, etc.
- APIs para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme, Qt etc.

SQL

- Implementa SQL92/SQL99.
- Llaves primarias (primary keys) y foráneas (foreign keys).
- Columnas auto-incrementales.
- Índices compuestos, únicos, parciales y funcionales en cualquiera de los métodos de almacenamiento disponibles, B-tree, R-tree, hash ó GiST.
- Eventos LISTEN/NOTIFY.
- Sub-selects, Joins, Vistas (views), Disparadores (triggers), Reglas (Rules), Herencia de tablas (Inheritance), etc.

2.3.3. Framework

Claramente, no implementamos el sistema en diferentes frameworks para saber cual era más eficiente, pero hicimos nuestra elección basados en referencias encontradas en internet y por el conocimiento que teníamos sobre algunos.

Existen muchos análisis comparativos entre frameworks que podemos consultar. Algunos de los consultados se realizaron un par de años atrás, por lo que no son válidos para las nuevas versiones de las herramientas. Otros no modelan nuestro escenario de trabajo al no ocupar PostgreSQL como motor de Base de Datos.

	CakePHP	Symfony	Zend Framework	Code Igniter
Durabilidad	80,00 %	80,00 %	90,00 %	63,33 %
Solución industrializada	50,00 %	67,65 %	64,71 %	47,06 %
Adaptabilidad	83,33 %	100,00 %	100,00 %	83,33 %
Estrategia	50,00 %	50,00 %	41,67 %	41,67 %
Promedio	65,83 %	74,41 %	74,10 %	58,85 %

Tabla 2: Rendimiento de frameworks

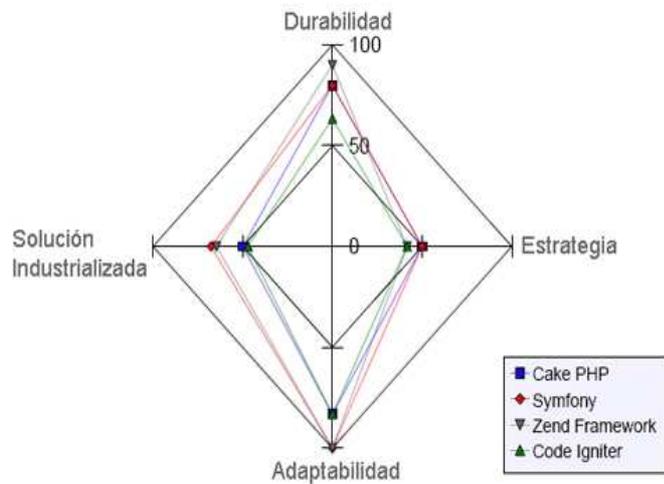


Figura 2: Rendimiento de frameworks

Analizando las comparaciones revisadas [25] [26] [27] [28] [29] se observa que, si bien CakePHP no es ostensiblemente más rápido que sus pares, llegando incluso a ser más lento en algunas pruebas, podemos sacrificar cierta eficiencia por el excelente manejo con la base de datos. Las características adicionales aportan velocidad y orden a la programación.

En la Tabla 2 y la Figura 2, se muestra un estudio que resume algunas pruebas y características de algunos frameworks, incluido CakePHP.

Otra razón por la que se escogió, es por que en Verbux existe un gran manejo de este framework, ya que desde hace cerca de un año todos los desarrollos de aplicaciones han sido implementados ocupando esta herramienta. Particularmente el memorista posee un vasto conocimiento en el desarrollo de software ocupando esta herramienta lo que

aumentaría la productividad de la construcción del software, eliminando el tiempo de aprendizaje.

CakePHP es un framework de desarrollo de aplicaciones web escrito en PHP, creado sobre los conceptos de Ruby on Rails [16].

CakePHP empezó en 2005, cuando Ruby On Rails estaba ganando popularidad. Desde entonces la comunidad creció y generó muchos subproyectos.

Características Al igual que Ruby On Rails, CakePHP facilita al usuario la interacción con la base de datos mediante el uso de ActiveRecord. Además hace uso del patrón Modelo Vista Controlador. Sus principales características son:

- Compatible con PHP4 y PHP5.
- CRUD de la base de datos integrado.
- URLs amigables.
- Sistema de plantillas rápido y flexible.
- Helpers para AJAX, Javascript, HTML, formularios y más.
- Trabaja en cualquier subdirectorio del sitio.
- Validación integrada.
- Scaffolding de las aplicaciones.
- Listas de control de accesos (Access Control Lists , ACL).
- Sanitización de datos.
- Componentes de seguridad y sesión.

Aplicaciones de consola. CakePHP ofrece un CLI, o interfaz de línea de comandos (Command-line interface en inglés), accedendo a través de la consola por el comando 'cake'. Hay tres aplicaciones por consola disponibles en la biblioteca: 'bake' para la generación de modelos, controladores y vistas; 'acl' para la gestión de listas de control de acceso y, 'schema' para la creación del esquema de la base de datos y la migración.

Los desarrolladores también pueden crear sus propios ‘scripts’ y compartir funcionalidades a través de estos ‘tasks’ (en la jerga CakePHP). Los scripts tienen acceso a los modelos y controladores de la aplicación. Un ejemplo de uso para este caso, sería la actualización programada de los datos de aplicación de un RSS, con toda la potencia de la aplicación lógica de las relaciones y los datos creados con CakePHP.

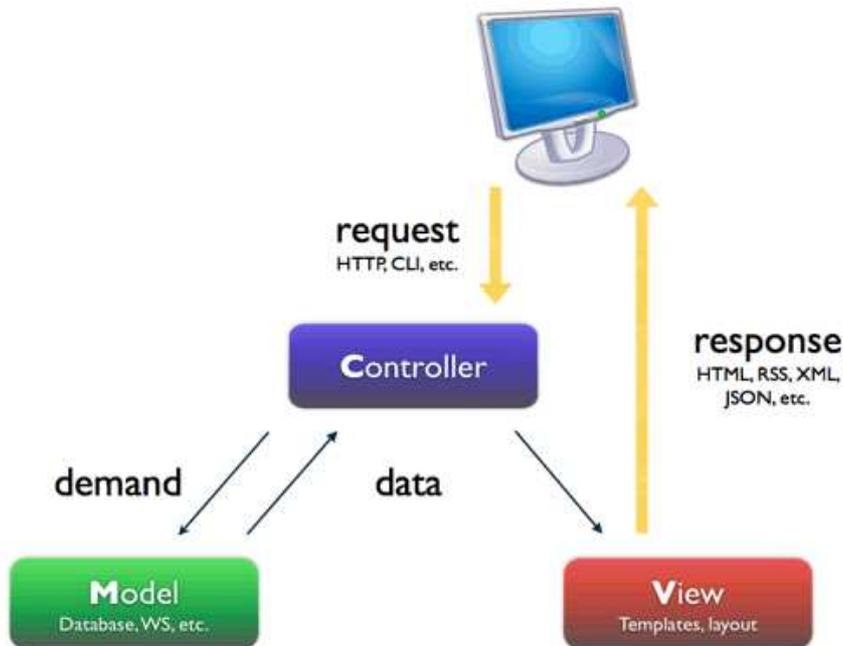


Figura 3: Modelo MVC

Patrón MVC CakePHP adhiere al patrón de arquitectura de software MVC [15] (Modelo Vista Controlador) 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 modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Descripción del patrón

- Modelo: esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos; por ejemplo, no permitiendo comprar un número de unidades negativo, calculando si hoy es el

cumpleaños del usuario, impuestos o importes de una bitácora para crear la factura.

- Vista: este módulo presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- Controlador: este módulo responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

3. Plan de Trabajo

3.1. Metodología

Para tener una visión completa del problema, se realizaron entrevistas a todos los usuarios finales del sistema (secretarias, técnicos, gerentes, etc.).

Los requerimientos obtenidos se validaron con el gerente a cargo de liderar este proyecto, acotando el posible alcance del software que se desarrolló.

En el desarrollo ocupamos una adaptación del modelo iterativo y creciente o incremental, la cual nos permitió tener nuevas funcionalidades en cada entrega.

3.2. Diseño

En esta etapa se elaboró el diseño y modelamiento de la base de datos, la cual satisfizo todos los requerimientos captados en la etapa anterior.

Para realizar una mejor especificación del sistema deseado, dependiendo del requerimiento, se implementaron prototipos del sistema para así tener un feedback por parte del gerente a cargo, en otros casos simplemente bastó con especificar los requerimientos y validarlos con el encargado.

Finalmente, se especificaron los distintos módulos y reglas lógicas del sistema en el documento final de requisitos con la información obtenida en esta etapa.

3.3. Construcción

La implementación del sistema se hizo de forma incremental y basándose en los requerimientos y documentación adquiridos en etapas anteriores.

Los entregables de este proyecto fueron modulares, entregando así en cada versión del sistema una herramienta funcional para el usuario.

Al gerente a cargo del proyecto le acomodó este tipo de avance, ya que no le importó tener todo el software funcionando en un principio, sino que necesitaba un programa que modularmente se fuera habilitando.

3.4. Validación

Las validaciones de los entregables se hicieron por parte del gerente a cargo, que a su vez aportó con una batería de datos de prueba para poder hacer el testeo de la aplicación.

Para validar la robustez del sistema, según lo analizado de las reuniones con la gerencia, el sistema se desarrolló con un framework de carácter libre.

La persistencia se logró una vez implementado completamente el sistema ya que no es necesario mantener la gran cantidad de documentos archivados en la oficina, sino que se tienen almacenados en la aplicación.

Como criterio de aceptación para la eficiencia del sistema sólo se necesitó un tiempo de respuesta adecuado para una aplicación de este tipo (unos cuantos segundos).

3.5. Integración

La integración con el sistema fue de vital importancia para cada incremento ya que se requiere un trabajo coordinado entre todos los módulos del sistema para poder lograr un mejor funcionamiento de la aplicación.

Para integrar los distintos incrementos fue necesario contar con un ambiente que simulase el ambiente final de producción, para así tener una mejor idea de como se comportaría el sistema en el futuro.

3.6. Validación del sistema

La validación del sistema es la última etapa antes de la puesta en marcha o en producción del sistema en Verbux. Se realizaron distintas pruebas para verificar que cumplía con los requisitos antes dispuestos.

4. Diseño del sistema

En esta sección especificaremos cada una de las etapas con las cuales se dió forma a la solución propuesta para Verbux.

4.1. Selección de personas a entrevistar

En esta sección se describen los roles de las personas de las cuales se obtuvieron los requerimientos, así como también, los distintos requisitos rescatados de estas entrevistas.

Para que las entrevistas tuvieran una visión transversal de la empresa, se entrevistó a personas con diferentes cargos dentro de Verbux. Estas entrevistas se efectuaron en la misma empresa y durante el horario de trabajo.

La primera persona que se entrevistó fue a Marco Rocco Guerrero, el cual en su calidad de Gerente de proyectos dió una visión del negocio a nivel de gestión, permitiendo así un punto de vista gerencial dentro de los entrevistados.

En segundo lugar, se entrevistó al Jefe de proyectos Juan Cuevas que, a su vez, fue Técnico de soporte a usuario, por lo que tiene conocimiento en ambas áreas, aportando a nuestras entrevistas una gama bastante amplia de visiones del negocio, importantes todas para su funcionamiento.

Luego se entrevistó a Mario Zamora quien es el Jefe de soporte al usuario de Verbux, lo que nos dió la posibilidad de lograr una mejor gestión en lo que respecta a recursos humanos.

Otra persona importante a la que se entrevistó fue la secretaria ejecutiva de Verbux, la cual pudo darnos una visión más detallada del funcionamiento de la empresa, y así especificar, aún más, los distintos requerimientos.

4.2. Especificación de requerimientos

Al comienzo de todo proyecto de desarrollo de software, es de vital importancia antes de desarrollar, la toma de requerimientos desde los usuarios. Lo ideal es que en esta etapa se clarifiquen los alcances y la forma de trabajar que se tendrá, para no generar falsas expectativas y evitar el rechazo del programa por parte del usuario.

Dentro del grupo de requerimientos obtenidos, se diferenciaron dos grupos, los funcionales y los no funcionales.

En este capítulo sólo se dará una descripción del requisito y su respectiva fuente. Un mayor detalle de esto se encuentra en el **Anexo A**.

4.2.1. Requisitos Funcionales

- Se necesita controlar la entrada y salida de los distintos técnicos y secretarias (MR, Marco Rocco).
- En caso de que un empleado de Verbox olvidase marcar la entrada o salida se deberá dar la opción para que el jefe de proyecto; del proyecto en qué trabajó, pueda cambiar las actividades de ese día para aquel trabajador (MR).
- Es indispensable cierto nivel de seguridad, tal que ninguna de las personas que ocuparán el sistema pueda ser suplantada (MR).
- Es necesario contar con la posibilidad de relacionar las horas trabajadas con las actividades que se realizan (MR).
- Se necesita contar con un detalle de las actividades que el técnico realiza diariamente (MZ, Mario Zamora).
- Todas las actividades que realiza el técnico deberán estar registradas entre las horas de entrada y salida de ese día (MZ).
- Se necesita controlar que todos los técnicos ingresen sus horas por lo menos semanalmente, de no ser así se debería gatillar un aviso (JC, Juan Cuevas).
- Los jefes de proyectos deberán tener la posibilidad de revisar las horas que se están asociando a sus proyectos con la posibilidad de objetarlas si así lo considerasen (JC).
- Es necesario para los jefe de proyectos contar con estadísticas sobre los proyectos bajo su responsabilidad, así como también poder realizar informes de los distintos indicadores de cada proyecto (JC, MR).
- Es necesario contar con la posibilidad de visualizar un resumen de todas las propiedades de cada proyecto, como por ejemplo, un resumen de las actividades que se le han asociado, o los recursos utilizados en él (MR, JC).

- Todos los proyectos son parte de un contrato entre Verbux y un cliente, los cuales cuentan con distintas restricciones de acción, como por ejemplo, horarios de horas extras, valores de las HH, extensión del proyecto, etc. Toda esta información deberá poder consultarse como parte del proyecto (MR, LG, Leslie González).
- El resumen mensual de las actividades realizadas en el proyecto es lo que se conoce en Verbux como la bitácora del proyecto, y sería deseable contar con una versión de ésta (MR, JC, LG).
- Idealmente la bitácora debe mostrar un resumen de los recursos invertidos en los distintos proyectos en los que se está trabajando (MR).
- Por distintas razones, algunas veces las horas cargadas no reflejarán las horas que deben aparecer en la bitácora, por lo que se debe dar la posibilidad al jefe de proyecto de que modifique las horas guardando ambos valores y horarios (JC).
- Se necesita poder administrar la información de los usuarios y los perfiles que se asocian a ellos. Esta posibilidad deberá estar disponible sólo para algunos usuarios del sistema, entre ellos los administradores y jefes de proyectos (MR).
- Es indispensable contar con la posibilidad de realizar informes con la información que se quiere guardar (MR, JC).
- Los usuarios podrán realizar informes con respecto a sus propios datos, tales como sus horarios de entradas y salidas, y las actividades que ha realizado durante el mes, etc. (MR).
- Otros informes sólo podrán ser realizados por los jefes de proyectos y los administradores del sistema (JC, MR).

4.2.2. Requisitos No Funcionales

- La navegación del sistema debe ser fácil para todos los usuarios (EP, Emilio Ponce).
- El diseño del sistema debe ser modular (MR).
- El sistema debe ser extensible (EP).
- El sistema debe correr en el servidor de Verbux, el cual es accesible desde Internet (MR).

- Todas las operaciones de los usuarios deben poder realizarse a través de los browsers con estándar W3C (MR).

4.3. Arquitectura Web

Se eligió desarrollar la aplicación como una aplicación web por las varias ventajas que da esta decisión. Se pueden enumerar las siguientes:

1. No necesita instalación previa, sólo se necesita tener un navegador que interprete HTML, W3C estándar.
2. Es de fácil actualización, ya que simplemente se debe actualizar el código en el servidor, a diferencia de las instalaciones stand-alone.
3. Alta escalabilidad . La principal ventaja de una aplicación distribuida bien diseñada es su crecimiento apropiado, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware.

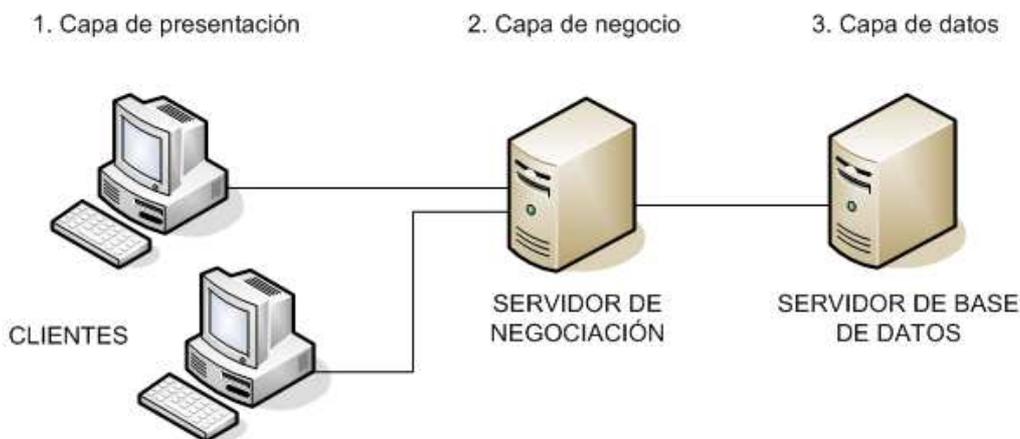


Figura 4: Arquitectura Web Tres Capas

4.4. Cakephp y MVC

Como se discutió en la Revisión Bibliográfica para realizar un desarrollo robusto y ágil es necesario contar con herramientas que lo permitan, así es como se eligió el framework Cakephp para la implementación de esta aplicación web, ya que cumple con todos los requisitos impuestos por la arquitectura web.

Cakephp cumple con el patrón de diseño MVC con el cual podremos independizar la vista del usuario, la capa de negocio de la aplicación y los datos del sistema.

Esta independencia trae consigo distintas ventajas sobre otros patrones de aplicación, como por ejemplo:

1. El desarrollo se puede llevar a cabo en varios niveles.
2. Desarrollos paralelos (en cada capa).
3. Aplicaciones más robustas debido al encapsulamiento.
4. En caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.
5. Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación monolítica).
6. Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad).

Finalmente, podemos decir que al construir la aplicación con CakePHP cumplimos con los objetivos que se plantearon en la toma de requerimientos, particularmente los requisitos de modularidad y extensibilidad, dado que el desarrollo con este framework nos ayuda a proporcionar ambas características.

4.5. Arquitectura de Datos

La arquitectura de datos del sistema debe ser capaz de poder responder las distintas preguntas que se presentan como requerimientos, y a su vez poder almacenar los datos entregados por los usuarios, con la debida granularidad.

La arquitectura de datos la podemos apoyar con distintos diagramas conceptuales que nos permitan una mejor concepción de lo que se tiene y de lo que se quiere lograr.

Para poder lograr esto, se necesita poder definir las distintas entidades de datos, cada una de las cuales tiene atributos y relaciones con otras entidades de datos. Esta etapa la podemos reducir en los siguientes cuatro pasos:

1. Debemos listar las entidades candidatas.

2. Definir las entidades, atributos y relaciones entre ellas.
3. Relacionar las entidades a las funciones del negocio.
4. Distribuir la arquitectura de datos con respecto a las distintas funcionalidades del sistema.

Podemos, después de esto, definir distintos grupos de entidades que nos serán útiles para el desarrollo de la aplicación, después de definir las podremos relacionar las distintas entidades para especificar finalmente el modelo de la base de datos capaz de satisfacer todos los requerimientos antes expuestos.

4.5.1. Personas

Las entidades que componen este grupo están orientadas a la administración y validación de los usuarios dentro del sistema. Cumplen con guardar todos los datos importantes de cada usuario.

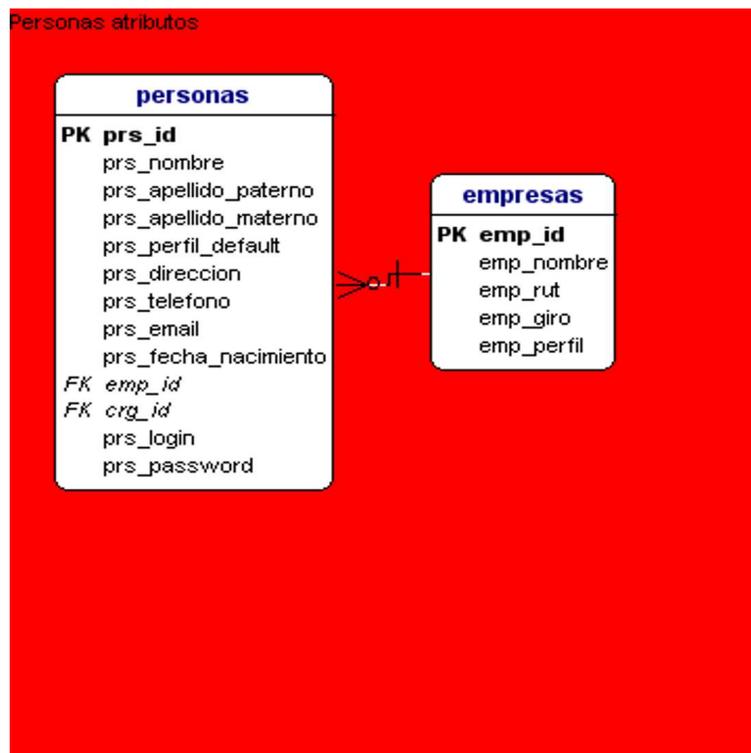


Figura 5: Grupo de entidades Personas

La entidad PERSONA fue modelada para poder guardar tanto los datos de información las personas como los datos de validación para el ingreso al sistema. La entidad EMPRESA esta referida a las distintas empresas a las que Verbux le presta servicios, aunque, en la lista de empresas también debería estar la propia Verbux.

La inclusión de la entidad EMPRESA está dada por dos motivos principalmente, el primero para identificar de qué empresa es cada proyecto y el segundo saber a qué empresa pertenece cada persona.

La relación PERSONA-EMPRESA es principalmente un dato que se utilizaría para la expansión del sistema, ya que está contemplado dar la posibilidad a los clientes de visualizar como van sus proyectos.

4.5.2. Proyectos

Este grupo de entidades tiene como objetivo principal describir y modelar los proyectos que Verbux tiene a su cargo, incluyendo los propios proyectos, proyectos que tienen como cliente el propio Verbux.

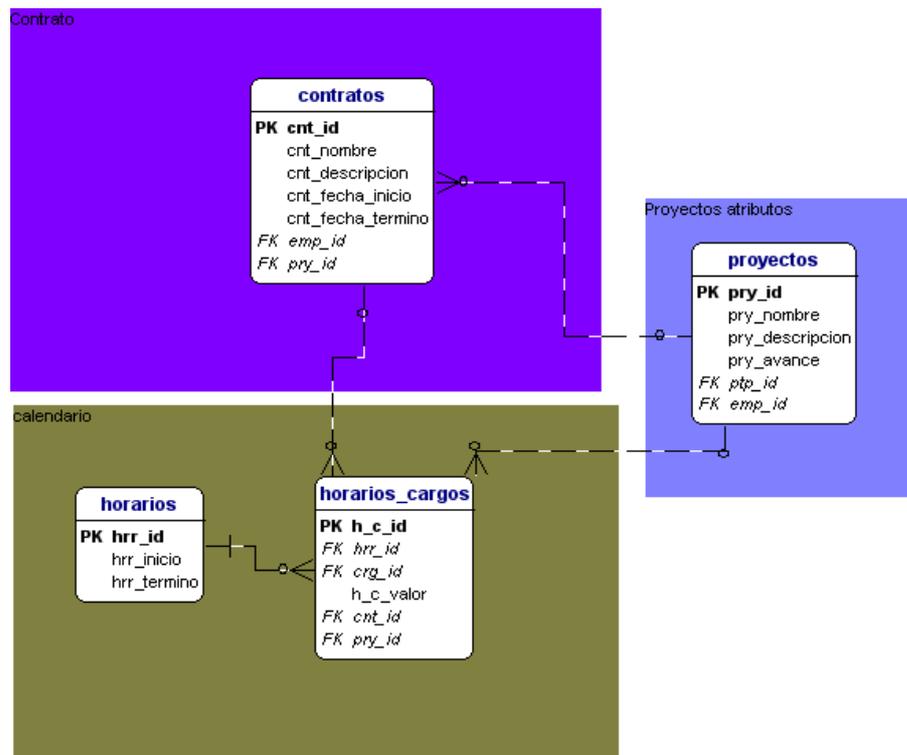


Figura 6: Grupo de entidades Proyectos

La entidad PROYECTO define cada proyecto que Verbux ha hecho o está ejecutando, con el respectivo modelamiento del contrato asociado al proyecto y los distintos tipos de horarios en los cuales se distribuirán horas.

PROYECTO tiene como atributos el nombre del proyecto (`pry_nombre`), una descripción de qué se trata (`pry_descripcion`), un indicador de cual es el avance del proyecto (`pry_nombre`), un identificador único para realizar búsquedas más fácilmente y dos relaciones externas que definen el tipo del proyecto y la empresa a la cual se le está prestando el servicio.

El tipo del proyecto se asigna desde una lista que está en constante aumento ya que Verbux tiene un área de Investigación y Desarrollo (I + D) que agrega nuevos tipos de proyectos a la lista.

La lista de empresas es otro grupo que cambia constantemente ya que el departamento comercial siempre está en búsqueda de nuevos clientes.

La entidad CONTRATO sirve principalmente para modelar los distintos horarios y restricciones que se imponen en cada proyecto para el cliente, sus fechas de inicio y término (`cnt_fecha_inicio` y `cnt_fecha_termino`), también le podemos definir un nombre al contrato y una breve descripción, ya que un mismo proyecto puede contener más de un contrato, e incluso ser parte de distintas empresas.

Finalmente, el grupo de entidades de horario está diseñada para poder modelar los distintos tipos de horas en las cuales se distribuyen las actividades realizadas por los técnicos e ingenieros de Verbux. Es necesario tener distintos horarios por distintos contratos ya que algunos de ellos cubren ciertas horas de soporte tanto al usuario como de administración de servidores.

El grupo está formado por dos entidades HORARIOS y HORARIOS_CARGOS las cuales tienen cada una objetivos muy específicos, la primera tiene como misión dar la granularidad de la distribución de horas, la segunda tiene como objetivo definir que cargo tendrán distintos valores de horas en diferentes horarios.

Por otro lado, las relaciones entre las entidades tienen dos objetivos, el primero es poder tener conexiones entre las distintas entidades que nos permitan asociar la información guardada, y en algunos casos las relaciones las ocupamos para así tener una mayor agilidad en las consultas a la base de datos.

4.5.3. Distribución y Aprobación

Este grupo principalmente está pensado para poder contabilizar las horas y actividades que los técnicos e ingenieros de Verbox han estado realizando y en qué proyectos se ha estado trabajando. También en este grupo se define la entidad que posibilita el marcado de entradas y salidas.

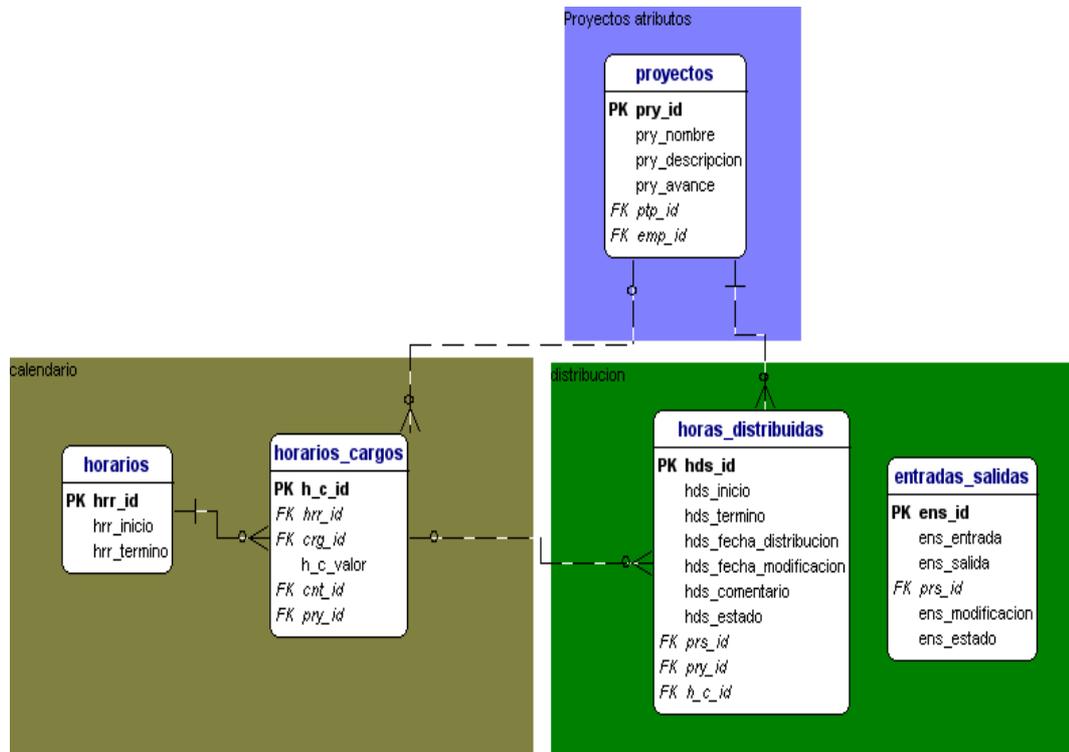


Figura 7: Grupo de entidades de distribución y aprobación

Principalmente este grupo está formado por dos tablas ENTRADAS_SALIDAS y HORAS_DISTRIBUIDAS, junto con las relaciones a otras tablas que ya se han mencionado (PROYECTOS y HORARIOS_CARGOS). En este caso, dichas tablas también se incluyeron para poder explicar mejor el funcionamiento del grupo.

La primera entidad, ENTRADAS_SALIDAS está diseñada para poder registrar las horas de ingreso y salida de los usuarios; tiene como principales campos los que registran la fecha, hora de ingreso y salida (`ens_entrada` y `ens_salida` respectivamente).

Esta entidad sólo se relaciona con PERSONAS ya que sólo es necesario saber de quién es el registro de entrada o salida.

La segunda entidad, HORAS_DISTRIBUIDAS, está pensada de forma que se puedan almacenar las distintas actividades en las que se distribuyen las horas que se obtienen de la tabla ENTRADAS_SALIDAS. Se puede también asociar cada registro con el tipo de horario en que se realizó la actividad guardada. Esto se hace por medio de la relación que existe entre las tablas HORAS_DISTRIBUIDAS y HORARIOS_CARGOS.

Junto con esto, la entidad HORAS_DISTRIBUIDAS tiene relaciones con otras entidades para que la información guardada ahí sea de utilidad, por ejemplo tiene relación con PERSONAS y PROYECTOS, lo cual nos permite saber quién trabajó en qué proyecto, si a esto se suma la relación mencionada en el párrafo anterior podremos responder la pregunta, qué persona trabajó en qué proyecto y entre qué horarios.

Por último, esta entidad cuenta con un campo para poder hacer una descripción de lo que se hizo (hds_comentario) y un atributo que define el estado de cada registro (hds_estado), por ejemplo, si éste está en vías de aprobarse, si está rechazado o definitivamente sí se aprobó.

4.5.4. Ingresos y Egresos

Este grupo ayudará a estimar, comparar y evaluar los proyectos en cuanto al aspecto presupuestario, ya que se podrá tener una correlación entre los que se ha gastado y lo que se presupuestó como gasto. Esta relación es de suma importancia, ya que es una de las principales preguntas que desde un principio la gerencia de Verbux quería responder.

Este grupo está formado por dos pares de entidades, las primeras guardan la información proyectada de los proyectos y las otras guardan la información real que se recoge desde la información que los jefes de proyectos cargan en el sistema.

Las entidades que guardan la información proyectada dividen los gastos de los ingresos, la primera, GASTOS_PROYECTADOS tiene como objetivo planificar los gastos que se tendrán durante el proyecto, cada registro dependerá de las entidades CONTRATOS, PROYECTOS y GASTOS_TIPO.

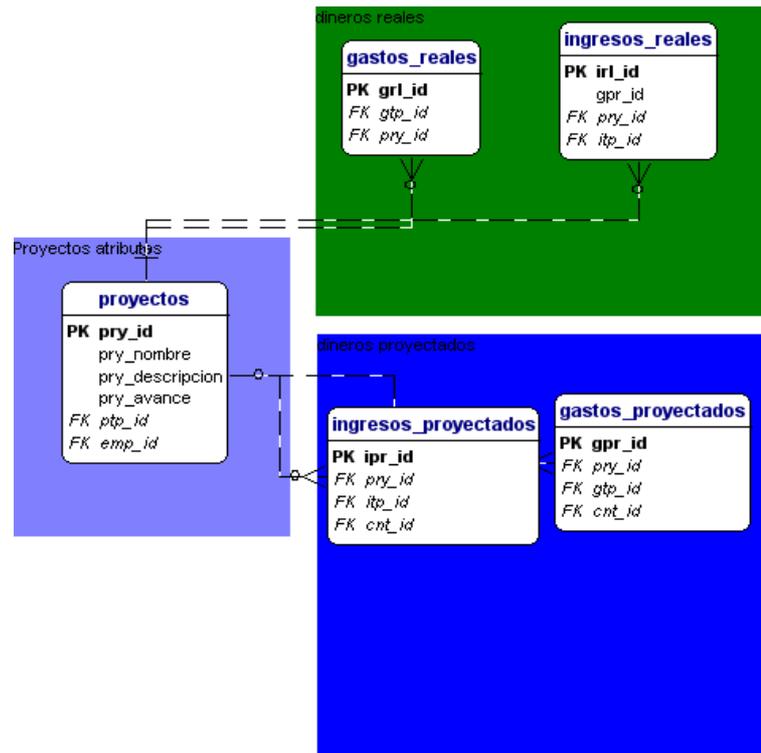


Figura 8: Grupo de entidades Ingresos y Egresos

Paralelamente la proyección de los ingresos se almacenará en una entidad llamada **INGRESOS_PROYECTADOS** la cual tiene relaciones con las entidades **CONTRATOS**, **PROYECTOS** y **INGRESOS_TIPO**, esta entidad está pensada para poder determinar cual sería la ganancia *a priori* del proyecto que estamos evaluando o proyectando.

4.5.5. Modelo Completo

En resumen, como lo muestra la figura, el resultado es un modelo que cumple con las restricciones antes expuestas y soluciona el problema de negocio de Verbux.

La entidad **CARGOS** está destinada a la clasificación de las personas dentro del cargo que ellas desarrollan en Verbux, sus atributos especifican cual será el nombre del cargo (**crge_nombre**), una pequeña descripción de lo que hace (**crge_descripcion**) y un valor de la HH asociada (**crge_valor_hora**).

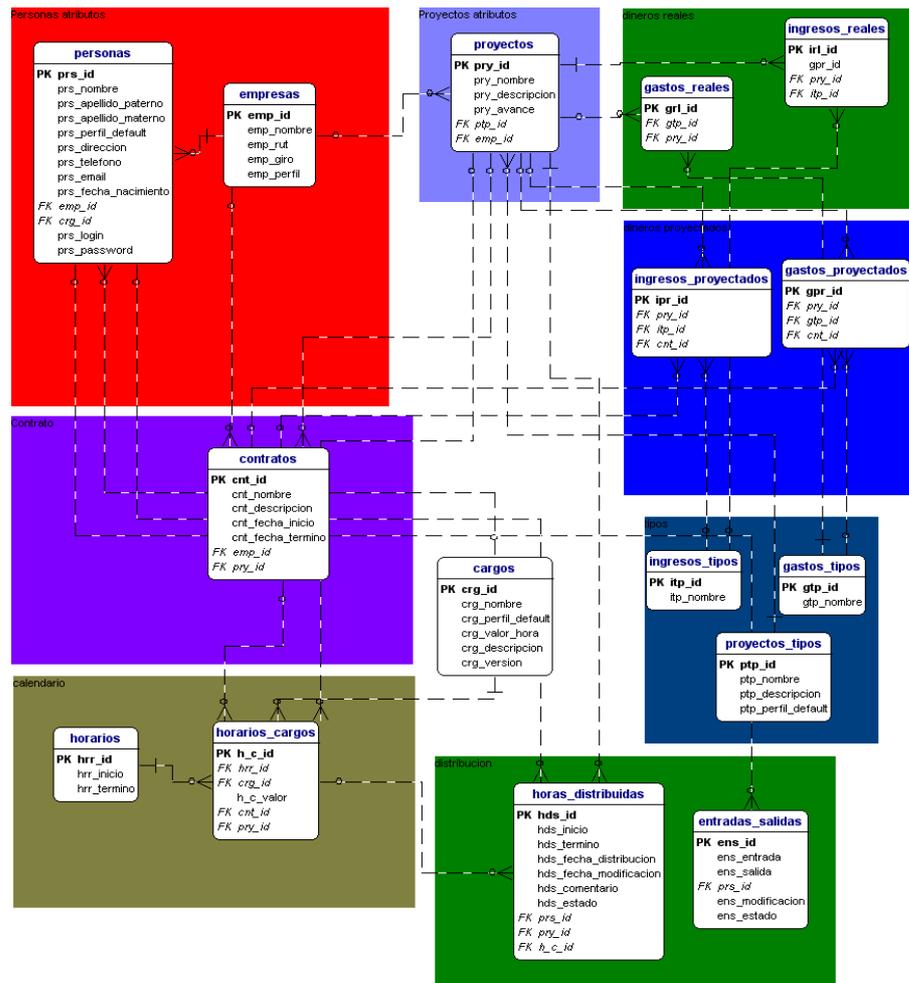


Figura 9: Modelo completo de entidades

Otras tablas omitidas en las explicaciones anteriores son INGRESOS-TIPOS y GASTOS-TIPOS, que cumplen la función de asociar a los distintos registros de gastos e ingresos, tanto proyectados como reales, un tipo que servirá para que posteriormente se puedan agrupar y contabilizar.

Por último, logramos armar un modelo de datos que satisface los requerimientos antes descritos y puede almacenar la información deseada para una posterior consulta de los registros y generación de informes.

5. Construcción del sistema

Con los antecedentes antes mostrados y las restricciones expuestas, se dará la mejor solución posible a las necesidades de la empresa.

Como se explicó en la revisión bibliográfica, algunas de las restricciones impuestas por la empresa impiden comprar un software de categoría como SAP. La implementación de un sistema PMS surgió como una buena opción pero luego de analizar sus posibilidades, se desestimó su implementación.

La adopción de un sistema libre también se consideró dentro de las posibilidades, pero se descartó por razones de rigidez.

Por lo que finalmente se decidió el desarrollo de un sistema ad-hoc para Verbux con herramientas de carácter libre para así tener justo lo que se necesita y tener la opción de ampliar el sistema si se necesitare en el futuro.

Para la implementación del modelo de datos se ocupará el sistema de gestión de bases de datos PostgreSQL, principalmente por su sencilla administración e implementa el estándar SQL.

Para el desarrollo, se eligió el framework CakePHP. Este combina la simplicidad de la generación automática de clases con la flexibilidad de la programación.

Este modelo de desarrollo nos permite independencia entre módulos. Junto con que el framework controla aspectos básicos del sistema como la configuración y parte de la navegación.

Al preocuparnos sólo de la parte medular del sistema (modelos, controladores y vistas), el diagrama de clase es bastante más simple que en otros casos. Ya que los modelos deben ser una traducción de lo que la arquitectura de datos nos entrega.

El diseño de las vistas tiene que ser coherente con las páginas de la empresa y contar con las herramientas necesarias de interacción con el usuario y los controladores deben aplicar las reglas del negocio al sistema, basándose en los requerimientos antes expuestos.

5.1. Roles

Dentro del sistema existen cuatro roles (administrador, jefe de proyecto, usuario y gerente), los cuales, permiten diferenciar los tipos de usuarios. La información de que rol o roles tiene cada usuario es conseguida desde el campo 'prs_perfil' el cual almacena un número correspondiente a los roles que posee el usuario en el sistema.

El rol usuario dentro del sistema, fuera de permitir el acceso a él, permite el ingreso al módulo de marcado de entradas y salidas (reloj) y distribución de horas en los proyectos e informes (sólo los relacionados con sus horas).

Todos los clientes del sistema deben tener a lo menos el rol de usuario asociado a su perfil, ya que de otra manera, el sistema no permitirá su ingreso.

El Rol administrador sólo tiene por objetivo mantener actualizado el sistema, sus listas desplegadas, su cartera de clientes, etc., así como también la lista de los usuarios del sistema. Este rol lo puede cumplir cualquier persona de la empresa, siendo asignado en primera instancia al gerente de Verbux.

Como jefe de proyecto se podrá tener acceso a los módulos de modificación de proyectos, aprobación de horas, informes, bitácora y modificación de gastos e ingresos.

Dentro de todos los módulos antes mencionados, el rol jefe de proyecto sólo podrá modificar la información de los proyectos que tiene bajo su responsabilidad, pero podrá visualizar los datos del resto de los proyectos.

Por último, el rol gerente podrá revisar, incluyendo permiso para actualizar, toda la información de los proyectos que se tiene almacenada y podrá generar reportes accediendo al módulo de informes.

5.2. Modelos

La construcción de los modelos en CakePHP es bastante simple, ya que ahí se declaran las relaciones que el sistema debe respetar entre las entidades, por lo que cada entidad antes descrita en la arquitectura de datos debe tener obligatoriamente una definición de su modelo.

Las relaciones mas importantes estan representadas en la Figura 10 con sus mutiplicidades y relaciones, las cuales debemos traspasar al tipo de modelamiento que CakePHP necesita.

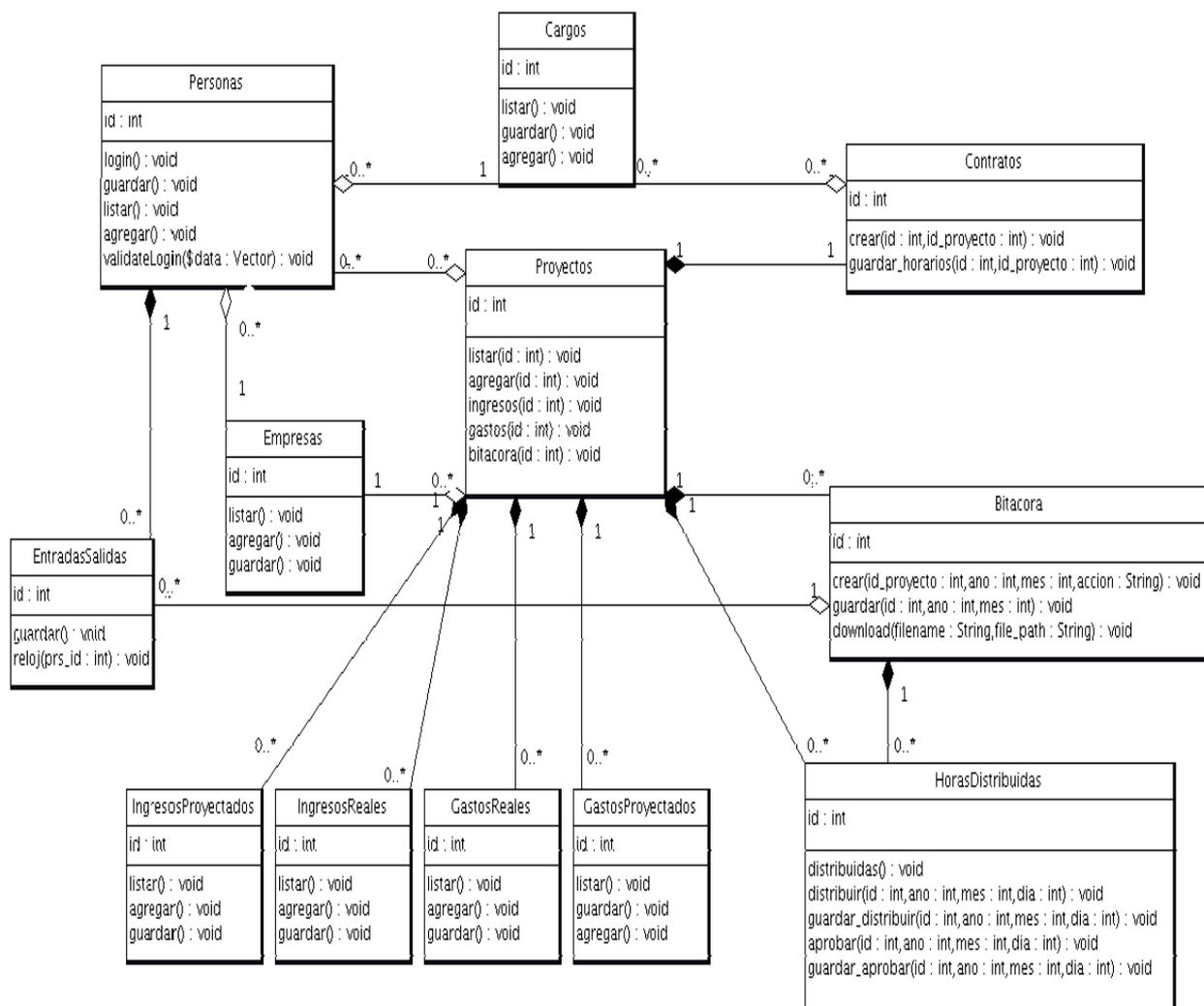


Figura 10: Diagrama de clases importantes

Al igual que en los modelos de datos, las relaciones tienen distintas características, y esto CakePHP lo maneja muy eficientemente declarando como atributos globales del modelo las relaciones.

Relación	Tipo de Asociación	Ejemplo
uno a uno	hasOne	Un usuario tiene una empresa.
uno a muchos	hasMany	Los usuarios en un sistema pueden tener entradas y salidas.
muchos a uno	belongsTo	Una entrada o salida pertenece a un usuario.
muchos a muchos	hasAndBelongsToMany	Los horarios tienen, y pertenecen, a contratos.

Tabla 3: Tipo de relaciones en CakePHP

Cada relación se debe especificar según lo muestra la Tabla 3, estas son la típicas relaciones en el modelamiento Entidad/Relación.

En los modelos, a su vez, se tiene la opción de declarar funciones las cuales serán visualizadas globalmente desde otros modelos y controladores que la incluyan.

El código se ve de esta manera:

```

/*Declaracion del modelo*/
class Persona extends AppModel {
/* Nombre con que se conocera el modelo*/
    var $name = 'Persona';
/* Nombre de la tabla relacionada al modelo en la BD*/
    var $useTable = 'personas';
/* Nombre de llave primaria */
    var $primaryKey = 'prs_id';
/* Declaracion de las relaciones HasMany*/
    var $hasMany = array(
        'EntradaSalida' => array (...
        )...
    );
/* Declaracion de las relaciones hasAndBelongsToMany*/
    var $hasAndBelongsToMany = array(
        'Proyectos' => array (...
        )...
    );
/* Declaracion de las funciones globales*/
    function validateLogin($data)
    {
        .....
    }
}

```

5.3. Vistas

Las vistas, en Cakephp, se manejan a través de ‘layouts’ que son propios de cada modelo y están ubicados en la carpeta ‘views’ del árbol de la aplicación.

En nuestro caso existirán dos clases de layouts, una para la interfaz de marcado de horas (Reloj), y otra para el resto de aplicación.

El primero sólo deberá contar con una cabecera, donde se visualizará el logo de la empresa, y un cuerpo, el cual deberá desplegar los campos necesarios para el marcado de entradas y salidas.

El segundo aparte de contar con la cabecera, deberá dividir la porción restante en dos, una para desplegar el menú personalizado de cada perfil y usuario del sistema, y otra para mostrar las vistas de los distintos módulos del sistema.



Figura 11: Layout del sistema

5.4. Controladores

Los controladores, en CakePHP, son los ‘directores de orquesta’ de las aplicaciones, ya que administran los recursos y ejecuta las acciones según sea el requerimiento. A diferencia de los modelos, no todos los controladores deben ser creados ya que en CakePHP, al crear los modelos crea instancias, las cuales tienen asociadas funciones predeterminadas de consultas y configuración.

Como en todo framework, es necesario seguir ciertas convenciones sobre nombres de archivos, funciones, clases y modelamiento de la Base de datos. Algunas de estas se pueden o no adoptarse. De hecho el modelamiento de la base de datos no cumple el estándar recomendado por CakePHP, pero de todas formas es recomendable seguir dichas convenciones.

Los nombres de archivos en los controladores deben ser en plural y llevar el postfijo ‘_controller’, para que el sistema los reconozca y los relacione con el llamado desde el navegador. En general los controladores tienen asociada una tabla en la base de datos. En nuestro caso existen dos controladores que no necesitan tablas, Bitácora e Informes.

Cada controlador es una instancia que recoge todos los datos de configuración y relacionales guardados en los modelos, vistas y otros archivos del framework. e declaran ocupando el nombre del archivo en formato CamelCase y extendiendo la clase ‘AppController’.

Al igual que en los modelos, podemos modificar ciertos parámetros que vienen configurados por defecto, como el nombre y el layout, también podemos incluir los modelos que se ocuparán en este controlador a través de la variable ‘uses’.

Las funciones definidas en el controlador son acciones que se ejecutan a través del navegador, cada acción tiene asociada una vista con el mismo nombre de la función, la cual despliega los resultados del controlador al ejecutar el llamado.

```
/*Declaracion del controlador*/
class ProyectosController extends AppController {
/*Nombre del controlador*/
    var $name = 'Proyectos';
/*Componentes que se ocuparan dentro del controlador*/
    var $components = array('Auth', 'RequestHandler');
/*Helpers que se ocuparan dentro del controlador*/
```

```
var $helpers = array('Html', 'Form', 'Session', 'DatePicker');  
/* Modelos que se ocuparan dentro del controlador */  
var $uses = array('Proyecto', 'Persona', ....);  
/* Este controlador lista todos los proyectos */  
function index() {  
    /* esta funcion se ejecuta al llamar desde la url '//servidor/  
       proyectos/index' */  
}
```

CakePHP no incorpora nativamente un módulo de generación de archivos PDF, por lo que se le debe incluir. Para esto se incorpora la clase ezPDF [24]. Esto se hace a través de la característica ‘vendors’ del framework que permite incluir clases dentro de las mismas funciones.

Una vez incluida, podremos crear objetos de la clase ezPDF, con lo cuales debemos diagramar los informes de forma legible e inteligente, para que tenga un valor real para la empresa.

6. Descripción del sistema

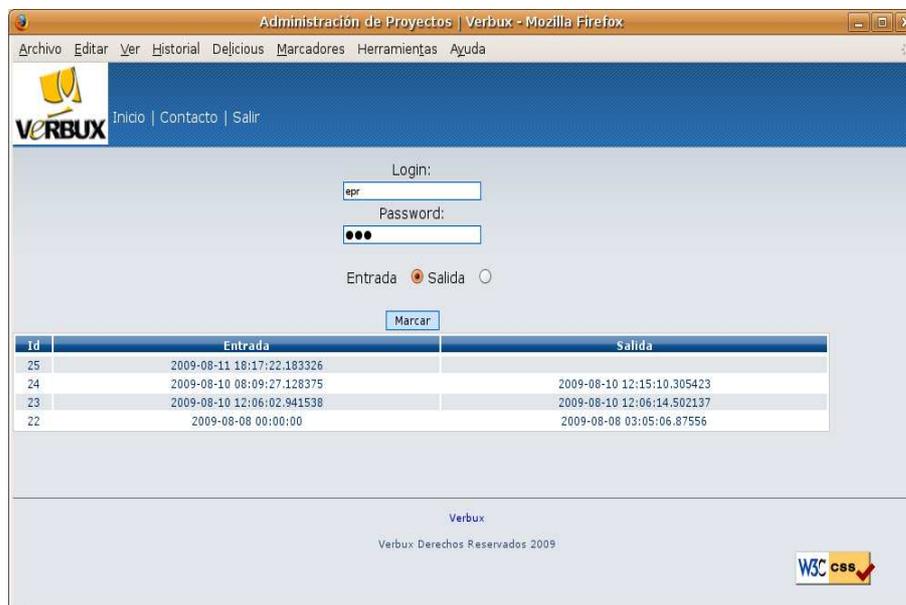
Se detallará el contenido y las funcionalidades que desarrolla cada módulo dentro del sistema. También se explicitará los pasos que deben seguir los usuarios para realizar las distintas acciones.

El sistema cuenta con dos interfaces, una que sólo permite el marcado de las horas de ingreso y egresos de los usuarios a Verbux, y otra que permite el registro de las distintas funcionalidades del sistema.

6.1. Interfaz de marcado de horas (Reloj)

Esta interfaz cuenta con tres casillas una para el ingreso del login, otra para el password y un selectbox en el cual se especifica si se está marcando una entrada o una salida. El usuario deberá llenar los 3 casilleros para poder marcar una entrada o salida de Verbux.

En caso de olvido de algún marcado, el reloj marcará como no trabajado ese día y el usuario tendrá que pedir una autorización para poder solucionar el problema a través de un módulo corrector habilitado especialmente para estas situaciones.



Id	Entrada	Salida
25	2009-08-11 18:17:22.183326	
24	2009-08-10 08:09:27.128375	2009-08-10 12:15:10.305423
23	2009-08-10 12:06:02.941538	2009-08-10 12:06:14.502137
22	2009-08-08 00:00:00	2009-08-08 03:05:06.87556

Figura 12: Interfaz de marcado de horas

El reloj podrá ser visible desde internet, y no se restringirá a la red de Verbux solamente, esto ya que en algunas ocasiones los técnicos pasan directamente a la empresa en la cual se está trabajando.

6.2. Interfaz del sistema

Esta interfaz cuenta a su vez con varios módulos los cuales cumplen distintas funciones y ,como se pidió en los requisitos, no todos son visibles para todos los tipos de perfiles.

6.2.1. Distribución de horas

Esta interfaz es visualizada por todos los usuarios y sirve para distribuir las horas que se cargan desde el reloj.

Cuenta con una interfaz en la cual se debe elegir la fecha, el rango de horas y el proyecto en el cual se quieren cargar horas.

Figura 13: Distribución de horas

Se debe adjuntar a su vez una descripción de lo que se hizo, dónde se hizo y quién recibió el trabajo para después llenar la bitácora.

El requerimiento del rango de horas, se debe a que de la distribución se sacará las horas que se deben cobrar al cliente, las cuales, para poder saber en que tipo de horario están (horario caro o barato), se necesita especificar la hora en que trabajó.

Con esto, el módulo de distribución de horas servirá como un sistema de control no sólo para saber si la gente está cumpliendo con sus horas, si no que también, como una posibilidad de ver cuanto tiempo libre está teniendo cada persona al mes.

Las horas restantes que las personas no puedan asignar a ningún proyecto, deberán asignarlas a un proyecto especial de Verbox con el cual se podrán obtener reportes, estadísticas y otros elementos de decisión útiles para los gerentes.

6.2.2. Aprobación de horas

Esta interfaz es visualizada sólo por los jefes de proyectos y tienen como objetivo poder validar las horas que los usuarios distribuyen.

Al igual que en el módulo de distribución de horas, la aprobación de horas se debe hacer en forma diaria, para ello se cuenta con un campo en el cual se selecciona la fecha que se quiere aprobar, demás esta decir que cada jefe de proyecto ve solamente información sobre los proyectos que tiene a su cargo.

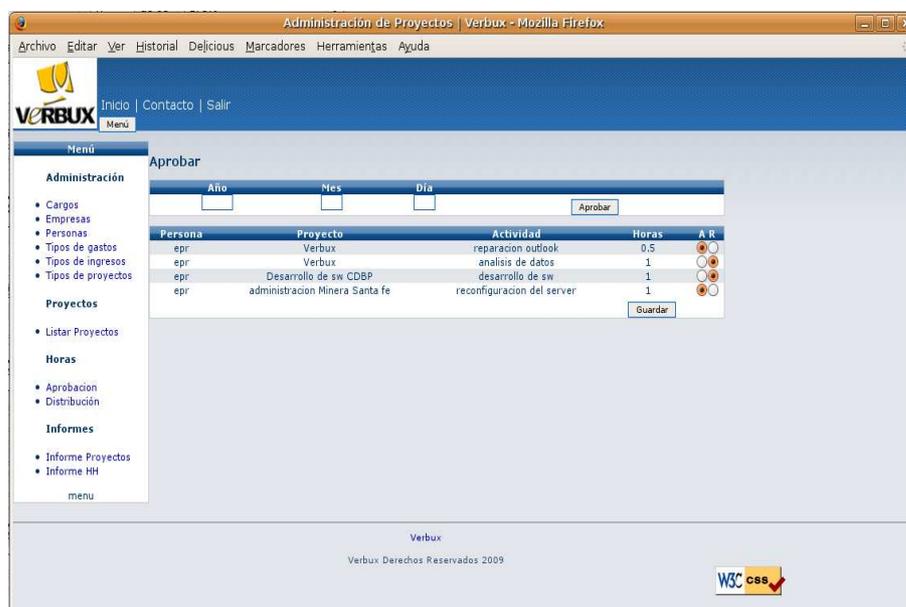


Figura 14: Aprobación de horas

La aprobación se hace simplemente chequeando la casilla de 'APROBADA', por defecto esta casilla debería estar chequeada, dejando la posibilidad de cambiarla a 'RECHAZADA'.

Las horas también se pueden rechazar, de ser así, aparece en la página de inicio del sistema para el usuario que las distribuyó tenga un aviso de que se le han rechazado.

El usuario podrá corregir las horas rechazadas o cargarlas nuevamente en alguna otra actividad.

6.2.3. Generación de bitácora

Esta interfaz es una de las más relevantes en todo el sistema, ya que acá se resumen todas las actividades que se realizaron en un proyecto durante los plazos establecidos, generalmente un mes.

The screenshot shows the 'Bitácora' interface in a Mozilla Firefox browser window. The page title is 'Administración de Proyectos | Verbox - Mozilla Firefox'. The interface includes a sidebar menu with categories like 'Administración', 'Proyectos', 'Horas', and 'Informes'. The main content area displays a table of activities for the month of August 2009.

Id	Fecha	Comentario	Usuario	Horas	Horario		Horas a cobrar	Estado		
					N	E		C	D	N
19	2009-08-11	analisis de datos	epr	1.5	<input checked="" type="radio"/>	<input type="radio"/>	1.5	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
13	2009-08-08	reparacion server	epr	1.5	<input checked="" type="radio"/>	<input type="radio"/>	34	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
15	2009-08-07	informe de registros	epr	1	<input type="radio"/>	<input checked="" type="radio"/>	33	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
26	2009-08-10	reparacion outlook	epr	0.5	<input checked="" type="radio"/>	<input type="radio"/>	0.5	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Totales				4.5	<input type="text" value="36"/>	<input type="text" value="33"/>	69			

Buttons: 'Cargar', 'Imprimir', 'Guardar'.

Figura 15: Bitácora

La generación de la bitácora consta de varias etapas, la primera es la elección del proyecto a generar, luego se debe seleccionar el período del cual se requiere la bitácora, posteriormente se deben seleccionar las horas que se le presentarán al cliente, teniendo la posibilidad de dejar algunas para meses siguientes o desecharlas simplemente, a su vez se debe elegir el tipo de profesional y el horario que se está cobrando, para que por último se genere la bitácora del proyecto.

Las primeras etapas son casi simultáneas, ya que se debe seleccionar el proyecto y el período dentro de la misma pantalla a través de una lista seleccionable.

La segunda etapa consiste en seleccionar las horas que serán parte de la bitácora de ese período, para esto se debe seleccionar el estado en que quedará el registro. Estos estados pueden ser:

1. Cobrable: esto quiere decir que en este período el registro se incluirá en la bitácora, y se deshabilitará para una próxima ocasión.
2. Nula: esta clasificación significa que la hora está correctamente asociada al proyecto, está aprobada por el jefe de proyecto pero no se cobrará al cliente por que puede haber sido un error por parte de Verbux y se asumirá el costo.
3. Demorada: esto significa que el registro de las horas está validado y corresponde al proyecto, incluso que se tiene que cobrar en algún momento pero se postergará el cobro, esto puede deberse a una opción del jefe de proyecto o a un acuerdo con el cliente.

Posteriormente sólo queda ver si las personas que realizaron el trabajo son las apropiadas, ya que en algunas ocasiones ingenieros deben realizar actividades de técnicos por razones de tiempo, ubicación y/o disponibilidad, pero al proyecto sólo se le debe asociar la actividad hecha por un técnico.

6.2.4. Informes

Esta interfaz está diseñada para generar reportes con los datos almacenados en el sistema, evidenciando lo que está sucediendo con los distintos proyectos y en la empresa en general.

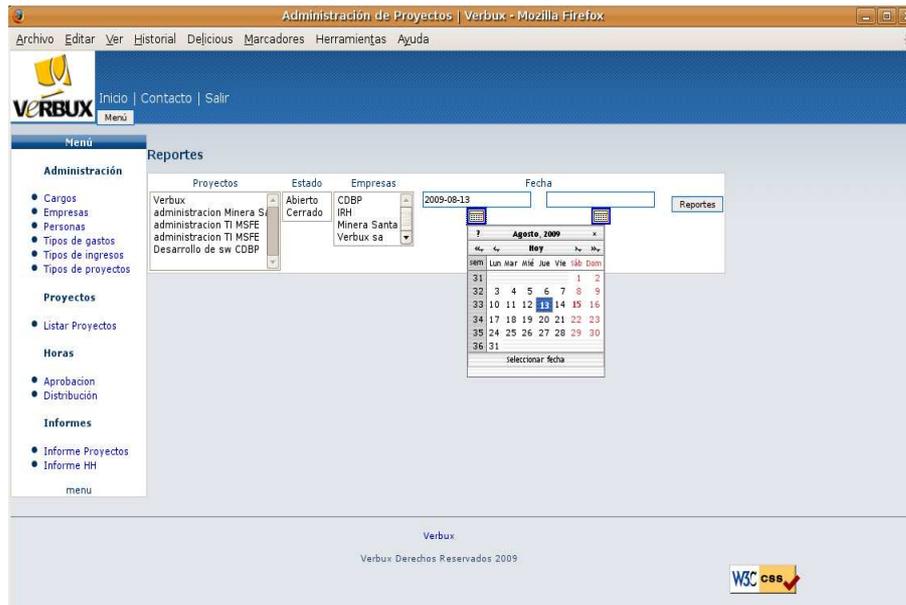


Figura 16: Informes libres

Se pueden realizar dos tipos de informes: los estructurados y los libres, los de tipo estructurados solamente se eligen los proyectos que se quieren informar y se descargan los informes, en cambio los libres tienen más tipos de filtros para poder realizar los reportes junto con más campos que se pueden mostrar.

Existe también un tipo de informe para todos los usuarios del sistema, el cual muestra el detalle de sus horas de entrada y salida, junto con un detalle de las actividades guardadas en el sistema.

6.2.5. Administración del sistema

Este módulo sirve para la mantención del sistema, por ejemplo crear usuarios, agregar cargos, etc.

La mantención del sistema está dada por la modificación de los datos almacenados en la aplicación, junto con el agregado.

Otra misión importante es administrar los perfiles de los distintos usuarios, ya que con esto podemos discriminar que usuarios cumplirán con los roles antes establecidos.

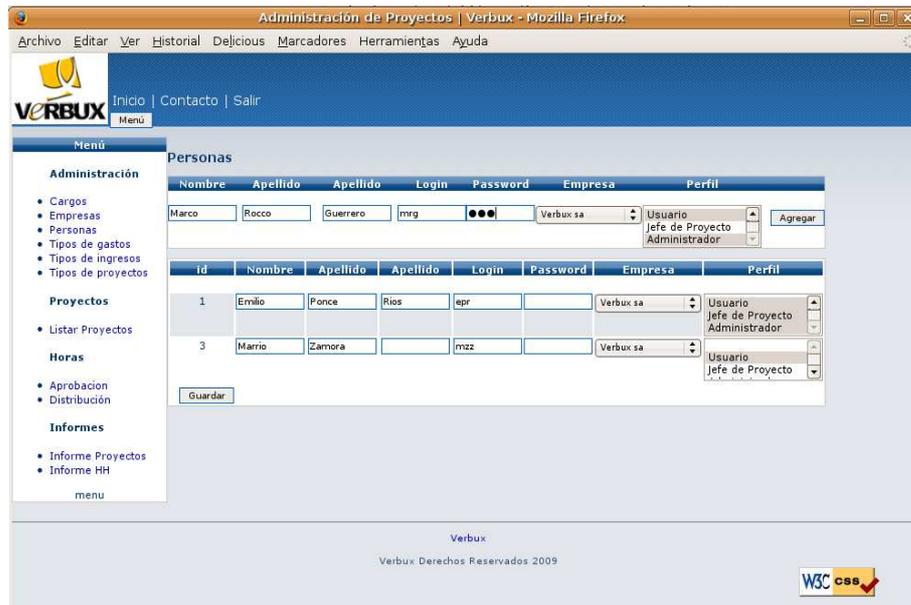


Figura 17: Administración de personas

Como se dijo anteriormente este módulo sólo podrá ser visto, por un usuario con perfil administrador, que en un principio sería la secretaria junto con el gerente de Verbox, esto es posible ya que una persona puede tener mas de un perfil asociado.

6.2.6. Creación y administracion de proyectos

Se usa para definir y crear los proyectos de los cuales se quiere controlar sus procesos, gastos, ingresos y actividades asociadas a él.

Este módulo es visualizable por los jefes de proyectos. Cada jefe solo tendrá acceso a modificar los proyectos a su cargo junto con poder revisar los datos e historial, de los proyectos pasados y en ejecución.

la creación de los proyectos puede realizarse con un contrato asociado o no, de ser así, se le asociarán los valores de los horarios de los distintos cargos relacionados al proyectos.

Por último, este módulo actuará ingresando datos al sistema, principalmente la información de los ingresos, gastos y horarios, permtiendo al jefe de proyecto controlar que la información que se asocie al proyecto bajo su mando sea coherente y útil para el resto de la empresa.

Administración de Proyectos | Verbug - Mozilla Firefox

Archivo Editar Ver Historial Delicioso Marcadores Herramientas Ayuda

VERBUG Inicio | Contacto | Salir
Menú

Ingresos >> Desarrollo de sw CDBP

Proyectados			
	Tipo	Descripción	Valor
3	Hardware	venta server	1500000
2	Administracion	contrato	1000000

Guardar

Reales			
	Tipo	Descripción	Valor
3	Hardware	server factura	1500000
2	Administracion	factura n°234	1238000

Guardar

Total Proyectados	Total Reales	Balance
2500000	2738000	238000

Verbug
Verbug Derechos Reservados 2009

WSC CSS

Figura 18: Agregar ingresos

6.3. Resultados

Los módulos anteriormente descritos nos proporcionan una serie de funcionalidades diseñadas para satisfacer los requerimientos enumerados en la etapa de especificación.

La correcta correspondencia entre módulos y requerimientos está dada por la matriz de trazabilidad, tanto para los requisitos de usuario (Anexo B) como para los requisitos de software (Anexo C).

Observamos que sólo un requisito de usuario (RU00005), no queda satisfecho por el desarrollo, principalmente por la prioridad asignada, tomando en cuenta que se tenía un tiempo acotado para el proyecto.

Administración de Proyectos | Verbug - Mozilla Firefox

Archivo Editar Ver Historial Delicios Marcadores Herramientas Ayuda

Inicio | Contacto | Salir

Menú

Administración

- Cargos
- Empresas
- Personas
- Tipos de gastos
- Tipos de ingresos
- Tipos de proyectos

Proyectos

- Listar Proyectos

Horas

- Aprobación
- Distribución

Informes

- Informe Proyectos
- Informe HH

menu

Nombre	Código	Descripción	Tipo	Empresa	Persona/Contrato
Desarrollo de sw CD	CDBP2009	desarrollo de software del sistema de administración de objetos de museo SURInternet	Desarrollo de software	CDBP	epi

Id	Nombre	Fecha	Código	Acciones
1	Verbug	2009-08-08		Gastos Ingresos Bitácora
2	administración Minera Santa fe	2009-08-08		Gastos Ingresos Bitácora
3	administración TI MSFE	2009-08-09	aTI-MSFE	Gastos Ingresos Bitácora
4	administración TI MSFE	2009-08-09	aTI-MSFE	Gastos Ingresos Bitácora
5	administración TI MSFE	2009-08-09	aTI-MSFE	Gastos Ingresos Bitácora

Verbug

Verbug Derechos Reservados 2009

W3C CSS

Figura 19: Módulo Proyectos

7. Conclusiones

7.1. Contribuciones

Este sistema entrega a Verbux un mejor control de los proyectos en ejecución, estadísticas de los proyectos que ya se han ejecutado y se están ejecutando, registro de los horarios de entrada y salida de los empleados de Verbux como de las actividades que realizan y se registra un menor tiempo en la elaboración de la bitácora por parte de los jefes de proyectos.

El control de proyectos se ve beneficiado al poder tener una visión general de la ejecución de los proyectos como por ejemplo el estado de avance, las horas a asociadas a él, balances generales sin necesidad de tener que sumar todos los gastos manualmente logrando así una estandarización en los procedimientos de Verbux.

Las estadísticas ayudan a poder realizar mejores estimaciones tanto de plazos como de costos para los proyectos que se estén evaluando, así como también, ayudan a los jefes de proyectos a no cometer los mismos errores de estimación en los cuales incurrieron ellos u otros anteriormente.

El registro de entradas y salidas determina las horas que el empleado de Verbux tiene disponible para realizar su trabajo y el control de actividades nos dice cuánto de ese tiempo es utilizado en actividades relacionadas con proyectos, su diferencia nos dice cuanto tiempo libre tiene cada empleado. Con esta información podemos realizar proyecciones de si se necesita más personal o si el tiempo libre que tienen se puede aprovechar capacitándolos de alguna manera.

El gran ahorro en tiempo, fuera de la generación de los balances parciales y totales, es la generación de la bitácora, ya que en un principio los jefes de proyectos debían recolectar todas las órdenes de trabajo, distribuirlas dentro de los distintos proyectos a su cargo. Luego se hacía el mismo filtro que hoy se hace por pantalla, elegir las horas que se cobrarán este mes, demorar o anular las otras.

En general, podemos decir que en algunos casos las actividades se ven suprimidas y en otras los tiempos disminuyen considerablemente. Por otro lado el número de proyectos que se maneja actualmente en Verbux ha aumentado, por lo que si no se hubiera dispuesto del nuevo sistema el número de HH sería mucho más grande que el que se usa actualmente.

Actividad	Sin Solución	Con Solución
Clasificación de OT	4 hrs aprox. (Secretaria)	No se necesita
Elaboración de bitácoras	2 hrs aprox. por proyecto (Jefe de proyecto)	Entre 20 y 30 min.
Balances	4 hrs. por proyecto	Instantáneo
Marcado de hora	En el libro de entrada	En libro de entrada y en el sistema
Monitoreo de proyectos	1.5 hrs aprox	Entre 15 y 20 min.
Información de proyectos pasados	2 hrs. y en algunos casos mucho más	Disponible en línea
Distribución de las horas diarias (por persona)	No se hacía; sólo se veían las OT	No más de 15 min. diarios

Tabla 4: Tiempos ahorrados

La clasificación de las OT (Ordenes de Trabajo), por ejemplo es una de las actividades que se ven suprimidas con la implementación de la solución, ya que si bien se tiene que guardar registro en papel de los trabajos realizados, lo que se debe tener actualizado ahora es la información que está en línea.

La elaboración de la bitácora ve disminuida su tiempo de creación en promedio una hora y media, por lo que mensualmente, con el número de proyectos actual, podemos ahorrar hasta treinta HH de jefe de proyecto.

Los balances por proyectos ahora son prácticamente instantáneos ya que están disponibles en los atributos del proyecto, lo que contrasta con lo que se tenía sin la solución. Estos demoraban cerca de cuatro horas lo que implicaba la recolección de gastos e ingresos junto con los distintos trabajos realizados.

El marcado de las entradas se ve afectado en un aumento del tiempo, ya que ahora aparte de firmar el libro de asistencia se debe ingresar la entrada o salida, contabilizando así el tiempo trabajado. Este aumento de tiempo es bastante menor, sólo algunos segundos.

La implementación de la distribución de horas es el cambio que conlleva la mayor inversión de tiempo, esto es entre diez y quince minutos diarios por empleado. Con

el tiempo, se espera que las personas se familiaricen con el sistema, y el tiempo por empleado no debería superar los cinco minutos por persona.

En la Tabla 4 vemos un resumen de estos cambios con sus respectivos tiempos. Muestra un ahorro considerable en tiempo gracias a la implementación de nuestra solución.

7.2. Trabajo Futuro

El sistema se construyó para que posteriormente se pudiese agregar nuevas funcionalidades fácilmente. Se pueden enumerar varias que serían útiles para el mejor funcionamiento de la empresa, como por ejemplo un módulo de facturación en base a la bitácora, acceso a los clientes para saber el estado de sus proyectos, ampliar y mejorar el módulo de informes estructurados, crear un módulo de estadísticas, mejorar el seguimiento de los requisitos, entre otros.

La adición del módulo de facturación ayudará a las secretarías a no cometer errores en la generación de las facturas ya que se harán directamente con los datos rescatados de la bitácora, de hecho, la intervención de ellas no será necesaria ya que sólo quedaría imprimir la factura.

Esto mejorará, a su vez, los registros de cobros hacia los clientes, evitando tener que revisar todas las facturas de un mes o incluso las de un año completo en busca de algún error, ya que sólo se tendría que consultar en el sistema para saber los datos reales.

El acceso a los clientes es una funcionalidad que tiene varias ventajas, por sobre todo, la transparencia de la empresa hacia ellos. La función de este módulo consiste en mostrar a los clientes el estado de avance de sus requerimientos y proyectos.

Otra funcionalidad un poco más elaborada, sería que el cliente tuviese acceso al detalle de la bitácora, esta funcionalidad puede ser bastante útil pero podría generar contradicciones con la bitácora final ya que algunas horas son desechadas y/o modificadas.

El módulo de informes estructurados cumple con los requerimientos que se establecieron en una primera iteración, pero es bien sabido que cuando se generan nuevas funcionalidades, por sobre todo de reportabilidad, los mismos usuarios adquieren nuevas necesidades como por ejemplo nuevos reportes u otros tipos de filtros para generarlos.

El seguimiento de los requerimientos se puede detallar aun más, especificando aún más las estadísticas que podremos obtener desde el nuevo modelo. Esta mejora está asociada a la visualización por parte de los clientes, podrán entregar un mayor detalle de lo que se está haciendo, el nivel de avance, y las estimaciones pertinentes.

La creación del módulo de estadísticas corresponde, al igual que el mejoramiento del módulo de informes, a una necesidad generada por el mismo sistema, ya que al tener la información almacenada de una forma estandarizada, se hace simple poder tener estadísticas de los clientes, proyectos, jefes de proyectos y la empresa, tanto en períodos como en general.

Referencias

- [1] Verbux. [2009]. '<http://www.verbux.com/>' [consulta: 02-04-2009]
- [2] CakePHP. [2009]. '<http://cakephp.org/>' [consulta: 02-04-2009]
- [3] Prado. [2009]. '<http://www.pradosoft.com/>' [consulta: 02-04-2009]
- [4] CodeIgniter. [2009]. '<http://codeigniter.com/>' [consulta: 02-04-2009]
- [5] ERP. [2009]. '<http://es.wikipedia.org/>' [consulta: 10-05-2009]
- [6] SAP. [2009]. '<http://www.sap.com/>' [consulta: 10-05-2009]
- [7] PMS. [2009]. '<http://es.wikipedia.org/>' [consulta: 08-08-2009]
- [8] SourceForge. [2009]. '<http://es.wikipedia.org/>' [consulta: 10-05-2009]
- [9] Openbravo. [2009]. '<http://www.openbravo.com/>' [consulta: 02-04-2009]
- [10] Lenguaje PHP. [2009]. '<http://www.php.net/>' [consulta: 01-05-2009]
- [11] 50 Extremely Useful PHP Tools. [2009]. '<http://www.smashingmagazine.com/2009/01/20/50-extremely-useful-php-tools/>' [consulta: 08-08-2009]
- [12] Lenguaje JAVA. [2009]. '<http://www.java.com/es/>' [consulta: 03-04-2009]
- [13] Lenguaje .NET. [2009]. '<http://www.microsoft.com/NET/>' [consulta: 03-04-2009]
- [14] Eclipse IDE. [2009]. '<http://www.eclipse.org/>' [consulta: 04-04-2009]
- [15] Patrón MVC. [2009]. '<http://es.wikipedia.org/>' [consulta: 02-04-2009]
- [16] Ruby on Rails. [2009]. '<http://es.wikipedia.org/>' [consulta: 04-04-2009]
- [17] MySQL versus PostgreSQL. [2009]. '<http://www-css.fnal.gov/dsg/external/freeware/pgsql-vs-mysql.html>' [consulta: 08-08-2009]
- [18] MySQL vs PostgreSQL. [2009]. 'http://www.wikivs.com/wiki/MySQL_vs_PostgreSQL' [consulta: 08-08-2009]
- [19] PostGreSQL vs. MySQL. [2009]. 'http://www.netpecos.org/docs/mysql_postgres/index.html' [consulta: 08-08-2009]
- [20] Database test. [2009]. '<http://tweakers.net/reviews/657/6>' [consulta: 08-08-2009]

- [21] MySQL. [2009]. '<http://www.mysql.com/>' [consulta: 04-04-2009]
- [22] PostgreSQL. [2009]. '<http://www.postgresql.org/>' [consulta: 04-04-2009]
- [23] Sobre PostgreSQL. [2009]. '<http://www.postgresql-es.org/>' [consulta: 04-04-2009]
- [24] ezPDF. [2009]. '<http://www.ros.co.nz/pdf/>' [consulta: 08-08-2009]
- [25] Comparativa entre CakePHP y Symfony. [2009]. '<http://aikon.com.ve/interesante-comparativa-entre-cakephp-y-symfony/>' [consulta: 10-06-2009]
- [26] Gráfica Comparación Frameworks. [2009]. '<http://lnx.googlewrite.com/chart.php>' [consulta: 10-06-2009]
- [27] PHP frameworks - Which one is Most Suitable for you?. [2009]. '<http://www.thecredence.com/php-frameworks-which-one-is-most-suitable-for-you/>' [consulta: 10-06-2009]
- [28] PHP Frameworks. [2007]. '<http://www.phpframeworks.com/>' [consulta: 10-08-2009]
- [29] Comparación y Rendimiento de Frameworks PHP. [2008]. '<http://codigolinea.com/2008/06/04/comparacion-y-rendimiento-de-frameworks-php/>' [consulta: 10-06-2009]

A. Requisitos

A.1. Requisitos de usuario

A.1.1. Requisitos funcionales

RU0001 - Controlar hora de salida y de llegada

Descripción : Se necesita de poder controlar la entrada y la salida de los distintos técnicos y secretarias para así poder contabilizar las horas trabajadas. En caso de que un empleado de Verbox olvidase marcar la entrada o salida se deberá dar la opción para que el jefe de proyecto del proyecto en que trabajó, pueda cambiar las actividades de ese día para aquel trabajador.

Fuente : Gerente

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : Técnico, secretaria, gerente

RU0002 - Evitar suplantación

Descripción : Es indispensable cierto nivel de seguridad tal que ninguna de las personas que ocuparán el sistema pueda ser suplantada.

Fuente : Gerente

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : Técnico, secretaria, gerente

RU0003 - Relación horas trabajadas

Descripción : Es necesario contar con la posibilidad de relacionar las horas trabajadas con las actividades que se realizan. Todas las actividades que realiza el técnico deberán estar registradas entre las horas de entrada y salida de ese día.

Fuente : Gerente

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : Técnico, secretaria, gerente

RU0004 - Control diario de actividades

Descripción : Se necesita contar con un detalle de las actividades que el técnico realiza diariamente.

Fuente : Gerente

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : Técnico, secretaria, gerente

RU0005 - Control de seguimiento sobre los técnicos

Descripción : Se necesita controlar que todos los técnicos ingresen sus horas por lo menos semanalmente, de no ser así se debería gatillar un aviso.

Fuente : Gerente

Prioridad : Baja

Estabilidad : transable

Fecha Actualización : 2009-06-18 12:03:00

Estado : No Cumple

Incremento : 2

Tipo : Funcional

T. Usuario Asociado : Técnico, secretaria, gerente

RU0006 - Revisión de horas asociadas a proyectos

Descripción : Los jefes de proyectos deberán tener la posibilidad de revisar las horas que se están asociando a sus proyectos con la posibilidad de objetarlas si así lo considerasen.

Fuente : Gerente

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : Técnico, secretaria, gerente

RU0007 - Estadísticas de proyectos

Descripción : Es necesario para los jefe de proyectos contar con estadísticas sobre los proyectos bajo su mando, así como también poder realizar informes de los distintos indicadores de cada proyecto.

Fuente : Gerente

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : Técnico, secretaria, gerente

RU0008 - Resumen del proyecto

Descripción : Es necesario contar con la posibilidad de visualizar un resumen de todas las propiedades de cada proyecto, como por ejemplo un resumen de las actividades que se le han asociado, o los recursos utilizados en él.

Fuente : Gerente

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : Técnico, secretaria, gerente

RU0009 - Contrato del proyecto

Descripción : Todos los proyectos son parte de un contrato entre Verbux y un cliente, los cuales cuentan con distintas restricciones de acción, como por ejemplo, horarios de horas extras, valores de las HH, extensión del proyecto, etc., Toda esta información deberá poder consultarse como parte del proyecto.

Fuente : Gerente

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : Técnico, secretaria, gerente

RU0010 - Bitácora del proyecto

Descripción : El resumen mensual de las actividades realizadas en el proyecto es lo que se conoce en Verbux como la bitácora del proyecto, y sería deseable contar con una versión de esta. Idealmente la bitácora debe mostrar un resumen de los recursos invertidos en los distintos proyectos en los que se está trabajando. Por distintas razones, algunas veces las horas cargadas no reflejarán las horas que deben aparecer en la bitácora, por lo que se debe dar la posibilidad al jefe de proyecto de que modifique las horas guardando ambos valores y horarios.

Fuente : Gerente

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : Técnico, secretaria, gerente

RU0011 - Administración del sistema

Descripción : Se necesita poder administrar la información de los usuarios y los perfiles que se asocian a ellos, esta posibilidad deberá estar disponible sólo para algunos usuarios del sistema, entre ellos los administradores y jefes de proyectos.

Fuente : Gerente

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : Técnico, secretaria, gerente

RU0012 - informes

Descripción	: Es indispensable contar con la posibilidad de realizar informes con la información que se quiere guardar. Los usuarios podrán realizar informes con respecto a sus propios datos, tales como sus horarios de entradas y salidas, y las actividades que ha realizado durante el mes, etc. Los jefes de proyectos y administradores del sistema Otros informes sólo podran ser realizados por los jefes de proyectos y los administradores del sistema.
Fuente	: Gerente
Prioridad	: Crítica
Estabilidad	: Intransable
Fecha Actualización	: 2009-06-18 12:03:00
Estado	: Cumple
Incremento	: 1
Tipo	: Funcional
T. Usuario Asociado	: Técnico, secretaria, gerente

A.1.2. Requisitos de calidad**RU0013 - Navegación fácil**

Descripción	: La navegación del sistema debe ser fácil para todos los usuarios.
Fuente	: Gerente
Prioridad	: Crítica
Estabilidad	: Intransable
Fecha Actualización	: 2009-06-18 12:03:00
Estado	: Cumple
Incremento	: 1
Tipo	: Calidad
T. Usuario Asociado	: Todos

RU0014 - Modularidad

Descripción : El diseño del sistema debe ser modular.
Fuente : Gerente
Prioridad : Crítica
Estabilidad : Intransable
Fecha Actualización : 2009-06-18 12:03:00
Estado : Cumple
Incremento : 1
Tipo : Calidad
T. Usuario Asociado : Todos

RU0015 - Código Extensible y de Calidad

Descripción : El sistema debe ser extensible.
Fuente : Gerente
Prioridad : Crítica
Estabilidad : Intransable
Fecha Actualización : 2009-06-18 12:03:00
Estado : Cumple
Incremento : 1
Tipo : Calidad
T. Usuario Asociado : Todos

A.1.3. Requisitos de Restricción**RU0016 - Ubicación del sistema**

Descripción : El sistema debe correr en el servidor de Verbux, el cual es accesible desde Internet.
Fuente : Gerente
Prioridad : Crítica
Estabilidad : Intransable
Fecha Actualización : 2009-06-18 12:03:00
Estado : Cumple
Incremento : 1
Tipo : Restricción
T. Usuario Asociado : Todos

RU0017 - Funcionalidad web

Descripción : Todas las operaciones de los usuarios deben poder realizarse a través de los browsers con estándar W3C.

Fuente : Gerente

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Restricción

T. Usuario Asociado : Todos

A.2. Requisitos de software

A.2.1. Requisitos funcionales

Estos requisitos indican cuáles deben ser las capacidades del software.

RS0001 - Ingresar Usuarios

Descripción	: El sistema debe proveer una interfaz que permita acceder al usuario al sitio, ingresando los datos correspondientes.
Fuente	: Desarrollador
Prioridad	: Crítica
Estabilidad	: Intransable
Fecha Actualización	: 2009-06-18 12:03:00
Estado	: Cumple
Incremento	: 1
Tipo	: Funcional
T. Usuario Asociado	: Todos

RS0002 - Creación y administración usuarios

Descripción	: El sistema debe proveer una interfaz que permita al administrador gestionar los distintos usuarios.
Fuente	: Desarrollador
Prioridad	: Crítica
Estabilidad	: Intransable
Fecha Actualización	: 2009-06-18 12:03:00
Estado	: No Cumple
Incremento	: 1
Tipo	: Funcional
T. Usuario Asociado	: Todos

RS0003 - Creación y administración de proyectos

Descripción : El sistema debe proveer una interfaz que permita crear un proyecto con todos sus datos correspondientes, a su vez se podrán asociar las distintas proyecciones de gastos y recursos esperados para dicho proyecto.

Fuente : Desarrollador

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : Todos

RS0004 - Marcar entradas y salidas

Descripción : El sistema debe proveer una interfaz que permita acceder al usuario a la posibilidad de marcar sus entradas y salidas.

Fuente : Desarrollador

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : Todos

RS0005 - Distribución y planificación de horas

Descripción : El sistema debe proveer una interfaz que permita acceder al usuario a la posibilidad de distribuir sus horas trabajadas y planificar sus tiempos.

Fuente : Desarrollador

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : Todos

RS0006 - Registros y gastos de proyectos

Descripción : El sistema debe proveer una interfaz que permita a los jefes de proyecto la posibilidad de registrar los gastos de los distintos proyectos que están dirigiendo.

Fuente : Desarrollador

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : jefe de proyecto

RS0007 - Bitácora de proyectos

Descripción : El sistema debe proveer una interfaz para poder visualizar la bitácora del proyecto según el mes que se elija.

Fuente : Desarrollador

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : jefe de proyecto

RS0008 - Informes

Descripción : El sistema debe proveer una interfaz para poder realizar informes sobre los datos almacenados de los distintos proyectos.

Fuente : Desarrollador

Prioridad : Crítica

Estabilidad : Intransable

Fecha Actualización : 2009-06-18 12:03:00

Estado : Cumple

Incremento : 1

Tipo : Funcional

T. Usuario Asociado : jefe de proyecto

B. Matriz de trazabilidad, requisitos de usuarios

	Reloj	Informes	Bitácora	Distribución	Aprobación	Proyectos	Administración
RU0001	X						
RU0002	X	X	X	X	X	X	X
RU0003				X			
RU0004		X					
RU0005							
RU0006		X					
RU0007					X	X	
RU0008		X					
RU0009						X	
RU0010			X				
RU0011							X
RU0012		X					
RU0013	X	X	X	X	X	X	X
RU0014	X	X	X	X	X	X	X
RU0015	X	X	X	X	X	X	X
RU0016	X	X	X	X	X	X	X
RU0017	X	X	X	X	X	X	X

Tabla 5: Matriz de trazabilidad, requisitos de usuario

C. Matriz de trazabilidad, requisitos de software

	Reloj	Informes	Bitácora	Distribución	Aprobación	Proyectos	Administración
RS0001							X
RS0002							X
RS0003						X	
RS0004	X						
RS0005				X	X	X	
RS0006			X			X	
RS0007			X				
RS0008		X					

Tabla 6: Matriz de trazabilidad, requisitos de software

D. Glosario

Outsourcing outsourcing o subcontratación es el proceso económico en el cual una empresa determinada mueve o destina los recursos orientados a cumplir ciertas tareas, a una empresa externa, por medio de un contrato.

Opensource open source o Código abierto es el término con el que se conoce al software distribuido y desarrollado libremente.

BOM bill of materials o lista de materiales , Se ocupa para saber la lista y cantidad de materiales se necesita para conseguir el producto final.

CRM de la sigla del término en inglés “Customer Relationship Management”, son Sistemas informáticos de apoyo a la gestión de las relaciones con los clientes, a la venta y al marketing.

CVS también conocido como Concurrent Versioning System, es una aplicación informática que implementa un sistema de control de versiones, mantiene el registro de todo el trabajo y los cambios en los ficheros que forman un proyecto (de programa) y permite que distintos desarrolladores colaboren.

SourceForge es un software de colaboración para la administración de desarrollos. Provee una portada para un amplio rango de servicios para el ciclo de vida de desarrollo de software e integra un amplio número de aplicaciones de software libre.

HTML siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web.

Framework en el desarrollo de software, es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado.

HH Abreviación de Hora-Hombre la cual representa el trabajo de un hombre por una hora

W3C World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web.

Layouts Término inglés que se utiliza para referirse a la disposición de los elementos en una composición, por ejemplo los de un sitio web .

Albaranes Los albaranes son comprobantes de ventas a credito, es el documento que se emite cuando la venta es a credito y anterior a la factura. en españa las facturas son siempre las mismas para cualquier venta

AJAX acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas.

Javascript es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

URL significa Uniform Resource Locator, es decir, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

CRUD En computación CRUD es el acrónimo de Crear, Obtener, Actualizar y Borrar (Create, Read, Update y Delete en inglés). Es usado para referirse a las funciones básicas en bases de datos o la capa de persistencia en un sistema de software

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor, es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

ACL Una Lista de Control de Acceso o ACL (del inglés, Access Control List) es un concepto de seguridad informática usado para fomentar la separación de privilegios.

ACID es un acrónimo de Atomicity, Consistency, Isolation and Durability: Atomicidad, Consistencia, Aislamiento y Durabilidad en español. En bases de datos se denomina ACID a un conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción.

CamelCase es un estilo de escritura que se aplica a frases o palabras compuestas. El nombre CamelCase se podría traducir como Mayúsculas/Minúsculas Camello, aunque no es correcto en todos los contextos ya que la palabra inglesa Case no tiene traducción literal

PDF acrónimo del inglés Portable Document Format, formato de documento portátil) es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems.

E. Comparativa de frameworks

MVC Indica si el framework viene con soporte incorporado para una configuración Modelo-Vista-Controlador.

Multiple DB's Indica si el framework es compatible con múltiples bases de datos sin agregar nada.

ORM Indica si el framework soporta mapeo object-record, generalmente una implementación de ActiveRecord.

DB Objects Indica si el framework incluye otros objetos de base de datos, como por ejemplo TableGateWay.

Templates Indica si el framework viene incorporado con un motor de template o plantillas.

Caching Indicates whether the framework includes a caching object or some way other way of caching. Caching: Indica si el framework incluye objetos para almacenar caché o alguna otra forma de almacenamiento.

Validation Indica si en el framework se ha incorporado un componente de validación o de filtrado.

Ajax Indica si el framework viene con soporte incorporado para Ajax.

Auth Module Indica si en el framework se ha incorporado un módulo para el manejo de la autenticación de usuarios.

Modules Indica si el framework posee otros módulos, como un analizador de fuentes RSS, módulo para la generación de archivos PDF o cualquier otro módulo útil.

EDP Event Driven Programming o Programación Orientada a Eventos.

PHP Framework	PHP4	PHP5	IMVC	Multiple DB's	ORM	DB Objects	Templates	Caching	Validation	Ajax	Auth Module	Modules	EDP
Atelos 	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
ash.MVC 	-	✓	✓	-	-	✓	✓	-	✓	-	✓	✓	-
CakePHP 	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-
CodeIgniter 	✓	✓	✓	✓	-	✓	✓	✓	✓	-	-	-	-
DIY 	-	✓	✓	-	✓	✓	✓	✓	-	✓	-	-	-
eZ Components 	-	✓	-	✓	-	✓	✓	✓	✓	-	-	-	-
Fusebox 	✓	✓	✓	✓	-	-	-	✓	-	✓	-	✓	-
PHP on TRAX 	-	✓	✓	✓	✓	✓	-	-	✓	✓	-	✓	-
PHPDevShell 	-	✓	-	-	-	-	✓	-	-	✓	✓	✓	-
PhpOpenbiz 	-	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	-	-
Prado 	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
QPHP 	✓	✓	✓	✓	-	✓	✓	-	✓	✓	✓	✓	✓
Seagull 	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
Symfony 	-	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-
WACT 	✓	✓	✓	✓	-	✓	✓	-	✓	-	-	✓	-
WASP 	-	✓	✓	-	-	✓	✓	-	✓	✓	✓	✓	-
Yii 	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Zend 	-	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-
ZooP 	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	-	-

Figura 20: Comparativa de frameworks PHP