

The logo for the Revista Cubana de Ciencias Informáticas (RCCI) features the letters 'RCCI' in a bold, blue, sans-serif font. The letters are contained within a white rectangular box with rounded corners. Below the box, there are faint, light blue horizontal lines.

Revista Cubana de Ciencias Informáticas

ISSN: 1994-1536

rcci@uci.cu

Universidad de las Ciencias Informáticas

Cuba

Mesa Reyes, Yuniór; Vázquez Ortiz, Yudisney
Espacio de comunicación e intercambio para la comunidad técnica cubana de
PostgreSQL.
Revista Cubana de Ciencias Informáticas, vol. 5, núm. 1, 2011, pp. 1-13
Universidad de las Ciencias Informáticas
Ciudad de la Habana, Cuba

Disponible en: <http://www.redalyc.org/articulo.oa?id=378343671002>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Tipo de artículo: Artículo Original
Temática: Sistemas de bases de datos
Recibido: 18/3/2011 | Aceptado: 21/4/2011 | Publicado: 29/9/2011

Espacio de comunicación e intercambio para la comunidad técnica cubana de PostgreSQL.

Space of communication and exchange for PostgreSQL's Technical Cuban Community.

Yunior Mesa Reyes^{1*} y Yudisney Vázquez Ortiz²

^{1*} Centro de Tecnologías de Gestión de Datos (DATEC)-Universidad de las Ciencias Informáticas (UCI), Cuba. Carretera a San Antonio de los Baños Km. 2 ½, Reparto Torrens, Boyeros, Ciudad de La Habana, Cuba. ymreyes@uci.cu

² Centro de Tecnologías de Gestión de Datos (DATEC)-Universidad de las Ciencias Informáticas (UCI), Cuba. Carretera a San Antonio de los Baños Km. 2 ½, Reparto Torrens, Boyeros, Ciudad de La Habana, Cuba. yvazquezo@uci.cu

Resumen: La mayoría de las aplicaciones de software hoy día tienen como soporte un sistema de gestión de bases de datos. Actualmente en Cuba se emplea una amplia gama de gestores comerciales, privativos y libres, en las aplicaciones de empresas, organismos e instituciones del país. Un grupo de especialistas y operadores de PostgreSQL crearon la Comunidad Técnica Cubana de PostgreSQL, con el principal objetivo de contribuir de manera activa y decisiva con la soberanía tecnológica potenciando el uso y explotación del gestor libre más avanzado del mundo. Dicho objetivo no ha podido ser acometido con la fuerza requerida por la falta de comunicación efectiva entre todos los miembros de la Comunidad. La presente investigación describe una propuesta para dar solución a dicho problema, el Portal para la Comunidad Técnica Cubana de PostgreSQL. La solución será el resultado de la utilización de Drupal como sistema de gestión de contenidos; de Gimp como herramienta para el diseño y PHP como lenguaje de programación para el desarrollo de los módulos que el Drupal no tenga implementados y que son necesarios para darle cumplimiento a los requerimientos del mismo.

Palabras clave: portal web, postgresql, sistema de gestión de contenidos.

Abstract: The majority of the applications of software today take as a support a system of management of databases. Nowadays in Cuba there is used a wide range of commercial, exclusive and free managers, in the applications of companies, organisms and institutions of the country. A group of specialists and operators of PostgreSQL created PostgreSQL's Technical Cuban Community, with the principal aim to contribute in an active

and decisive way with the technological sovereignty promoting the use and exploitation of the most advanced free manager of the world. The above mentioned aim could not have been attacked with the force needed by the lack of effective communication between all the members of the Community. The present investigation describes an offer to give solution to the above mentioned problem, the Portal for PostgreSQL's Technical Cuban Community. The solution will be the result of Drupal's utilization as system of management of contents; of Gimp like tool for the design and PHP like language of programming for the development of the modules that the Drupal does not have implemented and that are necessary to give him fulfillment to the requirements of the same one.

Keywords: web portal, PostgreSQL, system of management of contents.

1. Introducción

La mayoría de las aplicaciones de software hoy día tienen como soporte un sistema de gestión de bases de datos, software de propósito general que facilita el proceso de definir, construir y manipular bases de datos para diversas aplicaciones. (ELMASRI, *et al.*, 2000). Existen varios sistemas comerciales que pueden emplearse para la gestión de los datos de las aplicaciones que se desarrollen, su elección depende de las necesidades y condiciones que tenga el cliente de dicha aplicación. La diversidad de gestores que se utilizan en el país conlleva a un grupo de inconvenientes, los principales son:

- Los gestores propietarios son utilizados de manera ilegal amparados en el bloqueo económico y comercial impuesto por Estados Unidos, de pagarlos los montos ascenderían a miles de dólares.
- Independientemente del dinero a pagar para utilizar los gestores propietarios, hay que contar con la alta posibilidad de que en estas soluciones existan puertas traseras por las que se escaparía información valiosa.
- Las empresas de desarrollo de software están imposibilitadas de comercializar productos que tengan como base gestores propietarios.
- Los gestores libres tienen un riesgo latente de que dejen de serlo en algún momento, como pasó con FoxPro y recientemente con MySQL.

Todas estas problemáticas contribuyeron a que un grupo de especialistas comenzaran a promover la utilización de PostgreSQL en las aplicaciones empresariales del país; alineados a que Cuba se encuentra hoy inmersa en un proceso de migración a software libre con el propósito de alcanzar una independencia tecnológica más que necesaria, sobre todo cuando de sistema de gestión de bases de datos se trata, teniendo en cuenta que la información siempre es sensible y más en un país bloqueado y asediado como lo es Cuba. Un grupo de especialistas y operadores de PostgreSQL crearon la Comunidad Técnica Cubana de PostgreSQL, con el

principal objetivo de contribuir de manera activa y decisiva con la soberanía tecnológica potenciando el uso y explotación del gestor libre más avanzado del mundo.

La Comunidad Técnica Cubana de PostgreSQL, a dos años de creada en la Universidad de las Ciencias Informáticas, cuenta ya con más de 450 miembros afiliados distribuidos en todas las provincias del país y el municipio especial Isla de la Juventud, con una mayor representación en las provincias de Ciudad de la Habana y Villa Clara donde se encuentran dos de los centros que acogen gran cantidad de los miembros.

Dentro de sus objetivos se encuentra: contribuir al fortalecimiento de la soberanía tecnológica cubana, desde el enfoque del desarrollo de tecnologías de bases de datos tomando como base a PostgreSQL; proveer soluciones integrales y consultorías relacionadas con la migración y la explotación de PostgreSQL y contribuir a la formación de especialistas de alto nivel en tecnologías de bases de datos. Por su parte los principales impactos que pretende lograr con su trabajo son: realizar aportes a los programas nacionales de informatización y migración a software libre; contribuir al desarrollo de una suite de herramientas para el soporte y la generalización de PostgreSQL y contribuir a la formación en tecnologías de bases de datos.

En Cuba, en ese entonces existía un portal para mantener actualizado a los seguidores de PostgreSQL, en sus inicios fungió como puente unificador entre los miembros de la comunidad dispersos por todo el país, y como punto de partida para lograr acciones de conjunto en el desarrollo del gestor. Dicho portal presentaba una estructura que no es la más adecuada para lograr los objetivos que se persiguen con este tipo de aplicaciones web, inicialmente no fue desarrollado con las herramientas propicias, por lo que se hace complejo llevar a cabo el aumento de servicios y prestaciones. La información que se mostraba en la aplicación no era suficiente para satisfacer las necesidades que requieren los miembros de la comunidad, ejemplo: el intercambio directo entre los usuarios, aumento de la información del tema tratado, un mayor nivel de servicios ya que las prestaciones que se ofrecían eran insuficientes. Lo expuesto anteriormente daba lugar a la existencia de problemas de comunicación entre los diferentes especialistas del tema, provocando un desconocimiento por parte de las demás personas acerca de lo que se realizaba, perdiendo tiempo muchas veces cuando se trabajaba sobre las bases de un trabajo que ya se encontraba desarrollado. Otra de las dificultades existentes es que todo aquel usuario sin acceso a la Internet no puede mantenerse actualizado de lo que mundialmente se lleva a cabo con respecto al gestor de base de datos PostgreSQL.

¿Por qué PostgreSQL? Por ser un sistema gestor de bases de datos objeto-relacional que soporta gran parte del estándar SQL; que cuenta con características avanzadas como consultas complejas, llaves foráneas, disparadores, vistas, integridad transaccional, control de concurrencia multiversión; que puede ser extendido por el usuario

añadiéndole tipos de datos, funciones, operadores, métodos de indexado y lenguajes procedurales; que funciona sobre 34 plataformas incluyendo Windows XP, Linux, FreeBSD, Solaris y Unix; que soporta gran cantidad de lenguajes para el desarrollo de las aplicaciones, como SQL, Java, Perl, Python, C, C++, Ruby y PHP; y que tiene además una comunidad de desarrollo y soporte detrás distribuida por todo el mundo. (POSTGRESQL, 2009). Con todas estas características queda claro que PostgreSQL es sin dudas el gestor de código abierto más avanzado del mundo.

Las restricciones del ancho de banda que tiene el país que imposibilita el acceso permanente a los repositorios de desarrollo del gestor, donde se encuentran las últimas versiones del mismo y la no existencia de un lugar donde guardar las diferentes versiones de instaladores del gestor, las herramientas en torno a él (replicación, balanceo de carga, diseño, clientes de administración) y los desarrollos y resultados de la propia Comunidad son elementos que de conjunto con los antes mencionados conducen a la imposibilidad de una comunicación frecuente y eficiente entre los miembros de la Comunidad Técnica Cubana de PostgreSQL y con la Comunidad Internacional de Desarrollo de PostgreSQL, dificultando el desempeño exitoso y el cumplimiento de las misiones de la misma.

2. Materiales empleados

Para el desarrollo del Portal se utilizarán las siguientes herramientas:

- **XP:** metodología de desarrollo ágil con un desarrollo iterativo e incremental. Se basa en pruebas unitarias, en la reutilización de código para lo cual se crean patrones o modelos estándares. Algunas de las particularidades más significativas de esta metodología es que propone la programación en pares o pareja y que el cliente forma parte del equipo de desarrollo, facilitando esto grandemente la comunicación entre todos los miembros del equipo del desarrollo y el cliente, permitiendo que el mismo esté en todo momento presente y de acuerdo con todos los cambios por los que pase el sistema en desarrollo (Extreme Programming: A gentle introduction).
- **Visual Paradigm:** suite de herramientas CASE (Ingeniería de Sistemas Asistida por Ordenador, sus siglas en inglés: Computer Aided Systems Engineering) profesionales que utiliza UML como lenguaje de modelado. Es un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación entre todos sus integrantes. (PRESSMAN,1998).

- **UML:** lenguaje que permite modelar, construir y documentar los elementos que forman un sistema de software orientado a objetos. Este ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. (UNIFIED MODELING LANGUAGE, 1999 - 2010)
- **Drupal:** sistema de gestión de contenidos de código abierto para la construcción de sitios web dinámicos; equipado con una potente combinación de funcionalidades y adaptabilidad, que puede soportar un rango diverso de proyectos web desde blogs personales hasta grandes sitios de comunidades. (DRUPAL, 2003)
- **Gimp:** programa de manipulación de imágenes GNU, distribución libre de software para tareas como el retocado, composición y creación de imágenes; es multiplataforma y multilenguaje. (GNU IMAGE MANIPULATION PROGRAM, 2010)
- **PHP:** lenguaje interpretado de propósito general, especialmente diseñado para el desarrollo web y puede ser embebido en HTML; utilizado para el desarrollo de los módulos que no provea el Drupal y que sean necesarios para dar cumplimiento a los requerimientos del mismo. (PHP, 2010)
- **Apache:** servidor Web que entre sus características fundamentales se pueden destacar que es flexible, rápido y eficiente, es multiplataforma y se desarrolla de forma abierta. Este está continuamente actualizado y evoluciona a mayor velocidad. Además, puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona. También incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.

3. Resultados y discusión

Para la confección del Portal se realizó un estudio de los diferentes portales de reconocidas comunidades de PostgreSQL como la Comunidad Internacional, la argentina, la brasileña, la peruana, la francesa y otras de las comunidades referenciadas en el portal de la Comunidad Internacional de PostgreSQL.

3.1 Requerimientos no funcionales del sistema

3.1.1 Requerimientos de Software

3.1.2 Para el cliente: Sistema Operativo Windows XP o superior y Linux, Navegador web Mozilla Firefox 3.0 o Internet Explorer 5 o versiones superiores.

3.1.3 Para el servidor: Sistema Operativo Windows Server, Linux (cualquier distribución), Servidor web: Apache 2, Lenguaje de Programación: PHP 5.3.

3.1.4 Para la Base de Datos: PostgreSQL 8.3 o superior.

3.1.5 Requerimientos de Hardware

3.1.6 Para el cliente: Procesador Pentium III, 128 MB de RAM, 100 MB de disco duro.

3.1.7 Para el servidor: Procesador Pentium IV a 3.0 GHz, 1 GB de RAM, 10 GB de disco duro.

3.1.8 Otros requisitos.

3.1.9 Para el desarrollo de la aplicación igualmente se identificaron un conjunto de requisitos de apariencia o interfaz externa, de seguridad, de soporte y de portabilidad que propiciarán un mejor desempeño y funcionamiento de la aplicación una vez concluida su implementación y puesta en marcha.

3.2 Planificación del proyecto

La metodología de desarrollo XP comienza con la fase de planificación. En esta fase los clientes plantean a grandes rasgos las historias de usuario, al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. En esta fase lo fundamental es la elaboración de las Historias de usuario, que son de interés para la primera entrega del producto.

3.2.1 Historias de Usuarios

Uno de los artefactos más importantes que genera la metodología XP son las Historias de Usuario. Éstas tienen similar propósito que los casos de uso y son confeccionadas por el cliente. Las mismas expresan su punto de vista en cuanto a las necesidades del sistema. Son descripciones cortas y escritas en el lenguaje del usuario sin terminología técnica que proporcionan los detalles sobre la estimación del riesgo y cuánto tiempo conllevará su implementación. A continuación se muestran 2 de las 36 historias de usuario definidas. El resto aparece en el Expediente del Proyecto.

Historia de Usuario	
Número : 2	Nombre: Gestionar Usuario
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo de desarrollo: Alto
Puntos Estimados: 1	Iteración asignada: 1
Programador responsable: Pedro Valentín Paneque Silva	
Descripción: Esta funcionalidad describe cómo se llevará a cabo el proceso de gestionar los usuarios que se autenticuen en el portal, donde el administrador podrá adicionar, eliminar, mostrar un determinado usuario y modificar sus datos. Datos de usuario: <ul style="list-style-type: none"> • Usuario (Obligatorio) • Dirección de correo electrónico (Obligatorio) • Contraseña (Obligatorio) • Confirmar contraseña (Obligatorio) • Estado de la cuenta • Idioma 	
Observaciones: Se deben llenar el todos los campos que sean obligatorios para que se registre el usuario.	

Figura 1. Historia de usuario

Historia de Usuario	
Número : 20	Nombre: Gestionar productos.
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo de desarrollo: Alto
Puntos Estimados: 0.5	Iteración asignada: 4
Programador responsable: Pedro Valentín Paneque Silva	
Descripción: Esta funcionalidad describe como se llevará a cabo el proceso para gestionar todos los productos sobre PostgreSQL, mediante esta funcionalidad el administrador del sistema podrá insertar, actualizar, mostrar y eliminar todos los productos que se brinden sobre el gestor, para su posterior descarga.	
Observaciones:	

Figura 2. Historia de usuario

3.2.2 Estimación de esfuerzo por Historia de Usuario

Otro punto dentro de la fase de planificación es la estimación del esfuerzo que costará implementar cada historia de usuario. La estimación del costo de la implementación de las Historias de Usuario es establecida por los programadores tomando como medida el punto. Un punto se considera como una semana ideal de trabajo, donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. Generalmente las historias de usuario valen de 1 a 3 puntos. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

3.3 Diseño del sistema.

Para el diseño de aplicaciones informáticas la metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML. En su lugar se usan otras técnicas como las tarjetas CRC (contenido, responsabilidad y colaboración). No obstante el uso de estos diagramas puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante.

3.3.1 Tarjetas CRC

Las tarjetas CRC permiten desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos. Las tarjetas CRC permiten que el equipo completo contribuya en la tarea del diseño. Para poder diseñar el sistema como un todo se debe cumplir con tres principios: Cargo o Clase, Responsabilidad y Colaboración (CRC).

- **Clase:** es cualquier persona, cosa, evento, concepto, pantalla o reporte.
- **Responsabilidades:** las responsabilidades de una clase son las cosas que conoce y las que realizan, sus atributos y métodos.
- **Colaboradores:** los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.

A continuación se muestran algunas tarjetas CRC para ejemplificar lo anterior. El resto aparece en el expediente del proyecto.

Tarjeta CRC	
Clase: Información	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> ✓ Gestionar información ✓ Realizar búsquedas ✓ Gestionar preguntas frecuentes ✓ Gestionar encuestas ✓ Realizar encuesta ✓ Gestionar solicitud ✓ Realizar solicitud ✓ Visualizar reportes estadísticos 	<ul style="list-style-type: none"> Usuario Eventos Servicios Foro Productos

Figura 4. Tarjeta CRC 6

Nombre de la clase.	
Responsabilidades	Colaboradores

Figura 3. Estructura de la tarjeta CRC

Tarjeta CRC	
Clase: Servicios	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> ✓ Mostrar servicios ✓ Inscripción de Usuarios en Cursos. ✓ Consultar noticias 	<ul style="list-style-type: none"> Eventos Usuario

Figura 5. Tarjeta CRC 4

3.3.2 Patrón arquitectónico

La arquitectura empleada para el diseño del portal web está basada en el patrón arquitectónico Modelo Vista Controlador (MVC). Este es un patrón 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 el controlador representa la Lógica de negocio. Este estilo se basa en separar el diseño de las aplicaciones en tres capas diferentes:

- **Modelo:** Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comprar un número de unidades negativo.

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

Este modelo tiene varias ventajas ya que con él se puede mostrar distintas variantes de interfaz gráfica simultáneamente, agregar nuevos tipos de vista no afecta el modelo y evita poner el código indebido en la capa impropia. También facilita el despliegue en caso de modificaciones en el modelo de datos. Permite la reutilización de los componentes y da facilidad para desarrollar prototipos rápidos. Otras de las ventajas es la sencillez para crear distintas representaciones de los mismos datos y la facilidad para la realización de pruebas unitarias de los componentes, así como de aplicar desarrollo guiado por pruebas. También la simplicidad en el mantenimiento de los sistemas y que los desarrollos suelen ser más escalables.

3.3.3 Diseño de la base de datos

El diseño de la base de datos es de gran importancia pues debe almacenar información y permitir a los usuarios recuperarla y actualizarla en base a sus peticiones. El modelo de datos describe la representación lógica y física de los datos persistentes en el sistema, es frecuentemente necesitado cuando se tiene un modelo de objetos y el mecanismo de almacenamiento se basa en una base de datos relacional.

El modelo de datos del portal tiene baja complejidad, debido a que está desarrollado en el CMS Drupal con un alto número de entidades manejadas por el sistema, por lo que no se contemplaran todas en el modelo.

3.3.4 Arquitectura de la información

Arquitectura de la Información (AI) es el conjunto de métodos y herramientas que permiten organizar los contenidos, para ser encontrados y utilizados por los usuarios, de manera simple y directa. La AI estará cumpliendo sus objetivos cuando un usuario entre por primera vez al sitio y pueda reconocer a quién pertenece el sitio web; lo pueda entender en forma rápida, sin esfuerzo y encontrar la información ofrecida fácilmente. Adicionalmente eso entregará el beneficio de que quienes producen el sitio podrán ubicar la nueva información sin tener que crear nuevas estructuras, y al mismo tiempo tendrán la libertad de incorporar nuevas iniciativas al sitio sin tener que partir de cero.

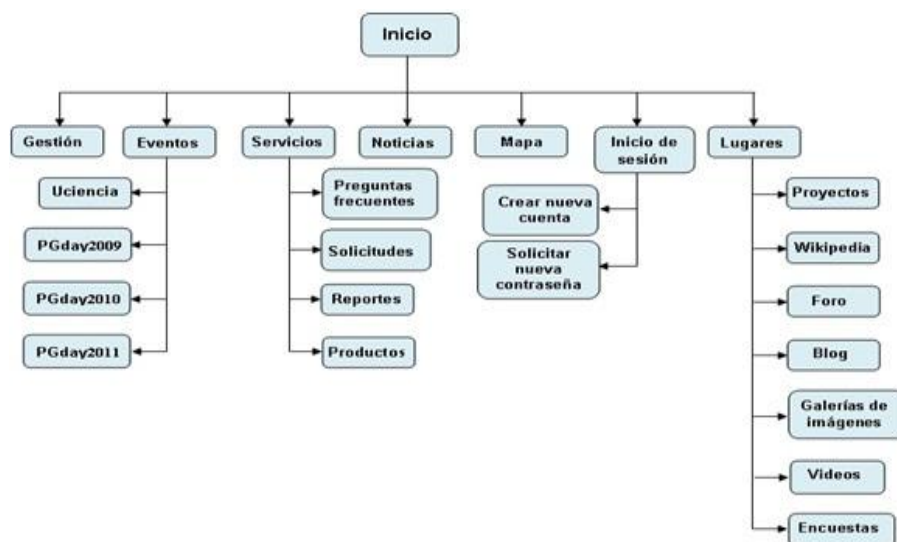


Figura 6. Mapa de Navegación

3.3.5 Diseño de la base de datos

El diseño de la base de datos es de gran importancia pues debe almacenar información y permitir a los usuarios recuperarla y actualizarla en base a sus peticiones. El modelo de datos describe la representación lógica y física de los datos persistentes en el sistema, es frecuentemente necesitado cuando se tiene un modelo de objetos y el mecanismo de almacenamiento se basa en una base de datos relacional.

El modelo de datos del portal tiene baja complejidad, debido a que está desarrollado en el CMS Drupal con un alto número de entidades manejadas por el sistema, por lo que no se contemplaran todas en el modelo.

3.4 Implementación y Prueba.

Según Kent Beck, en cualquier momento el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos. Por ello es que durante todo el desarrollo de esta metodología se debe haber diseñado la solución más simple que pueda funcionar para ser implementada en un momento determinado del proyecto donde la complejidad innecesaria y el código extra debe ser removido inmediatamente. En este momento se establecen varias iteraciones sobre el sistema antes de ser entregado. Algunos elementos que deben tomarse en cuenta de una iteración a otra son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

En XP, los programadores diseñan, programan y realizan las pruebas. El programador escribe las pruebas unitarias y produce el código del sistema. Para ello debe existir una comunicación y coordinación adecuada entre los programadores y el resto de los miembros del equipo. Toda la producción de código que se genera iteración tras iteración está dirigida por las pruebas unitarias, que son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse, que se obtienen a través de los Diseños de Casos de Prueba. A continuación se reflejan algunos de los desarrollados.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU8_P8	Nombre Historia de Usuario: Consultar eventos
Nombre de la persona que realiza la prueba: Raiza Ramírez Ramos	
Descripción de la Prueba: Prueba para la funcionalidad de consultar eventos	
Condiciones de Ejecución: El usuario seleccionará el evento que desea consultar.	
Entrada / Pasos de ejecución: El usuario escoge la funcionalidad evento y le aparecerán todos los eventos que están publicados en el sitio, donde el escogirá cual es el que desea consultar y entonces el sitio se re direccionará hacia esa página	
Resultado Esperado: El sistema muestra la página del evento escogido por el usuario	
Evaluación de la Prueba: Prueba Satisfactoria	

Figura 7. Diseño de Caso de Prueba 8

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU11_P11	Nombre Historia de Usuario: Inscripción en evento.
Nombre de la persona que realiza la prueba: Raiza Ramírez Ramos	
Descripción de la Prueba: Prueba para la funcionalidad de inscripción en el evento de participantes.	
Condiciones de Ejecución: El usuario para poder inscribirse en dicho evento debe estar autenticado.	
Entrada / Pasos de ejecución: El usuario escoge la funcionalidad inscripción del evento y le aparecerá la página de la inscripción donde deberá llenar los datos correspondientes, una vez llenados estos efectuará la inscripción. Datos: Nombre (obligatorio), Apellidos (obligatorio), Correo electrónico (obligatorio), Teléfono, Empresa o Entidad, País	
Resultado Esperado: El sistema realiza la inscripción.	
Evaluación de la Prueba: Prueba Satisfactoria	

Figura 8. Diseño de Caso de Prueba 11

Actualmente el Portal ya se encuentra liberado en su versión v1.0 por el Laboratorio Central de Calidad de la Universidad de las Ciencias Informáticas.



Figura 9. Página de inicio del Portal.

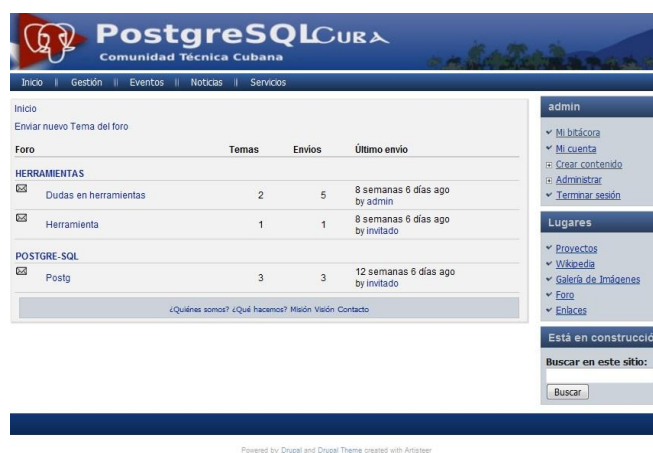


Figura 10. Funcionalidad Gestionar Foros implementado

3.1 Interfaces

La página de inicio del Portal tendrá una interfaz similar a la presentada en la figura que sigue, se muestran además algunas de las funcionalidades presentes en el mismo.

4. Conclusiones

El Portal para la Comunidad Técnica Cubana de PostgreSQL es implementado con el propósito de que los miembros de la Comunidad cuenten con un espacio que permita la comunicación efectiva entre ellos, manteniéndolos al tanto de los últimos desarrollos del y sobre el gestor; publicando las listas sobre él donde se aclaran dudas puntuales que tienen las personas con la utilización del mismo y creando un espacio para las descargas de las últimas versiones del gestor, las herramientas en torno a él (replicación, balanceo de carga, diseño, clientes de administración) y los desarrollos y resultados de la propia Comunidad. La implementación de los requerimientos del Portal es un arma indispensable para el correcto funcionamiento de la Comunidad y el logro de las misiones de la misma.

Referencias

dBI dataBased Intelligence. 2010. dBI. [En línea] 2010. [Fecha de consulta: 16 de febrero de 2010.] Disponible en: [http://store.databi.com/SearchResults.asp?Cat=1]

Drupal. 2003. Drupal. [En línea] 2003. [Fecha de consulta: 10 de mayo de 2010.] Disponible en: [http://drupal.org/project/drupal]

ELMASRI, RAMEZ Y NAVATHE, SHAMKANT B. 2000. Sistemas de bases de datos. Conceptos fundamentales. Tercera edición. s.l. : Editorial Addison-Wesley Iberoamericana, 2000.

GNU Image Manipulation Program. 2010. Gimp. [En línea] 2010. [Fecha de consulta: 10 de mayo de 2010.] Disponible en: [http://www.gimp.org/]

Microsoft corporation. 2010. Microsoft store. SQL Server Enterprise/Standar Edition. [En línea] 2010. [Fecha de consulta: 16 de febrero de 2010.] Disponible en: [http://store.microsoft.com/microsoft/SQL-Server-2008-Enterprise-Edition/product/E06A3697]

Microsoft Developer Network. 2010. Visual FoxPro Developer Center. Visual FoxPro 9.0. [En línea] 2010. [Fecha de consulta: 16 de febrero de 2010.] Disponible en: [http://msdn.microsoft.com/en-us/vfoxpro/bb190234.aspx]

Oracle corporation. 2009. Oracle Database New Features Guide, 11g Release 2 (11.2). 2009.

PHP. 2010. PHP. What is PHP. [En línea] 2010. [Fecha de consulta: 05 de mayo de 2010.] Disponible en: [\[http://php.net/\]](http://php.net/)

PostgreSQL Global Development Group. 2009. PostgreSQL 8.4.1 Documentation. California : s.n., 2009.