

# Proyecto para la Opción de Titulación por Trabajo Profesional: Diseño e implementación del sistema PEAT (Progressive Energy Audit Tool)

Alumno: Héctor Enrique Gómez Morales - No. de Cuenta: 401048742  
Tutor: Karla Ramírez Pulido

11 de septiembre de 2015

## 1. Introducción

Este proyecto aborda el diseño e implementación del sistema *Progressive Energy Audit Tool* (PEAT), el cual es un sistema web que permite a los usuarios de pequeñas y medianas empresas (PyMES) identificar y monitorear sus gastos de energía eléctrica, con el fin de obtener un mayor control en los costos de ésta.

Este sistema fue desarrollado siendo parte de la compañía **Software Next Door** en el cargo de **Senior Software Engineer**.

Las actividades que desarrollé durante este proyecto fueron:

- Implementación de la interfase de usuario.
- Implementación de una biblioteca que no solo permite realizar consultas al *backend* existente sino que permite la serialización/deserialización de objetos de Javascript a Ruby.
- Diseño e implementación de pruebas unitarias, funcionales y de integración.
- Implementación de un sistema de despliegue continuo de la aplicación, por medio de la automatización máxima de todos los procesos involucrados para lograr el despliegue del sistema.
- Implementación de optimizaciones tanto del lado de la arquitectura como del código desarrollado, para asegurar el soporte de por lo menos mil usuarios concurrentes.

## 2. Justificación

En una gran cantidad de proyectos resulta ser suficiente el hacer uso de un único lenguaje de programación para llevar a buen término el proyecto, es decir, terminarlo de

forma exitosa. Sin embargo conforme pasa el tiempo, las necesidades de las compañías y usuarios cambian, lo cual provoca que hagamos uso de una serie de tecnologías no previstas en un inicio, esto representa casi siempre el uso de más de un lenguaje de programación<sup>1</sup>.

En el proyecto de implementación del sistema PEAT cuenta con tres partes principales de desarrollo: (a) el *backend* implementado en su mayoría usando Javascript, con un núcleo escrito en Java, (b) una biblioteca en Ruby que permite la interacción del *backend* con aplicaciones web y (c) el *frontend* escrito con Ruby, dada sus ventajas para realizar sistemas web.

Durante este proyecto se implementó una biblioteca no solo para hacer uso de los servicios web ya existentes, sino para facilitar la creación de clases y objetos en Ruby basados en la jerarquía de clases definida en Javascript. Cabe mencionar que haciendo uso de una de las fortalezas de Ruby como lo es la metaprogramación para la creación de clases y sus atributos “al vuelo”, según las especificaciones enviadas por el servidor, se mantiene así la jerarquía de clases de un lenguaje a otro, lo cual fue uno de los objetivos cumplidos durante el proyecto.

Otro punto de creciente importancia es el despliegue de una aplicación, ya que por un lado hace algunos años las aplicaciones web se desarrollaban bajo el modelo de tres capas: la base de datos, el servidor de la aplicación y el servidor web; actualmente se tienen mas servicios que deben estar en línea sobre todo para garantizar un servicio concurrente; caches, equilibradores de carga, servidores de cola, etcétera. Por lo que automatizar el despliegue de una aplicación en sus diferentes contextos, desarrollo, producción y pruebas, es de vital importancia para el desarrollo de software en tiempo y en forma.

Durante este proyecto se hacen uso extensivo de técnicas como la metaprogramación y el despliegue continuo, para permitir un desarrollo acelerado, con el fin de obtener rápidamente retroalimentación del usuario final.

### 3. Descripción general del trabajo

Pacific Gas and Electric Company (PG&E) es una compañía proveedora de gas natural y electricidad, una de las más grandes compañías de Estados Unidos con sede en San Francisco, California. El sistema PEAT es el resultado de una licitación iniciada por PG&E para el desarrollo de un sistema enfocado en usuarios PyMES.

Su desarrollo es de importancia crítica tanto para PG&E, como para el gobierno de California puesto que su correcto funcionamiento es un requisito en una nueva ley de facturación de energía eléctrica desde el 2013. La nueva ley buscaba lograr distribuir la carga de la red eléctrica, siendo una de las formas principales para lograr esto el desincentivar el uso de la red eléctrica en horas pico al así darle la facultad a las compañías como PG&E de cobrar tasas mucho más altas en estas horas.

PG&E había realizado el despliegue de medidores inteligentes en la mayor parte de sus clientes, por lo que se tenía acceso a información muy detallada del consumo de energía de éstos.

Actualmente para darle el mayor valor e información a las empresas es necesario el obtener más datos de su entorno de operación; por ejemplo el número de edificios asociados a la cuenta, rubro de la empresa, número de empleados, etcétera. De esta forma, entre

---

<sup>1</sup>Debido a que ningún lenguaje de programación actualmente es el mejor en todos los contextos posibles de uso

mas detalles se pueda captar sobre la empresa el sistema tendrá más facilidad en obtener un desglose más robusto, completo y útil de sus consumos de energía.

El objetivo general es dar la mayor utilidad posible con la menor información disponible, fomentando en el usuario final el compartir dicha información sobre su empresa, obteniendo un mejor control acerca de su consumo energético.

Los objetivos secundarios que apoyan al objetivo general de este trabajo son:

- Implementación de una interfaz que permita la obtención de información del usuario de una forma eficaz y sencilla.
- Autenticar a los usuarios mediante el uso de credenciales de acceso obtenidas en el portal web de PG&E.
- Dar información útil aunque el usuario solo proporcione el mínimo de información sobre su empresa.
- Proporcionar recomendaciones para disminuir sus gastos en energía con base en el consumo e información proporcionada hasta el momento.
- Soportar por lo menos a mil usuarios concurrentes.
- Implementar la infraestructura para el despliegue continuo de la aplicación.

Cabe mencionar que la compañía C3 Energy, en la que trabajé por medio de Software Next Door, compañía ganadora de dicha licitación ya que contaba con la infraestructura necesaria para el procesamiento de una gran cantidad de datos de consumo de energía, tiene un sistema de monitoreo de consumo de energía enfocada a empresas de nivel multinacional. El reto era pasar de un sistema y procesos diseñados para una docena de clientes, a un sistema que diera servicio a cientos de miles de clientes PyMES.

## 4. Índice

### Introducción

#### 1. Situación actual

- 1.1. Contexto
- 1.2. Objetivo general
- 1.3. Objetivos secundarios
- 1.4. Propuesta de desarrollo

#### 2. Fundamentos teóricos para el desarrollo del sistema PEAT

- 2.1. Patrón Modelo-Vista-Controlador (MVC)
- 2.2. Servicios web RESTful
- 2.3. Desarrollo guiado por pruebas
- 2.4. Ruby, metaprogramación y lenguajes DSL

#### 3. Diseño e implementación del sistema PEAT

- 3.1. Arquitectura previa
- 3.2. Interfase de usuario
  - 3.2.1. Configuración inicial
  - 3.2.2. Tablero de energía
  - 3.2.3. Preguntas sobre el entorno de operación
  - 3.2.4. Lista de recomendaciones para el usuario
- 3.3. Biblioteca Bezel
  - 3.3.1. Arquitectura
  - 3.3.2. Metaprogramación
  - 3.3.3. Pruebas de rendimiento
- 3.4. Pruebas de integración
  - 3.4.1. Pruebas de conexiones concurrentes
  - 3.4.2. Pruebas de integración con PG&E
- 3.5. Optimizaciones y cache
  - 3.5.1. Balanceo de carga
  - 3.5.2. Configuración Nginx y Passenger
  - 3.5.3. Piscina de conexiones HTTP con *Passenger* y *Patron*
  - 3.5.4. Inicialización e invalidación de cache
- 3.6. Despliegue continuo
  - 3.6.1. Computo escalable en la nube (*Amazon Elastic Compute Cloud*)
  - 3.6.2. Herramientas para automatización (*RVM* y *Capistrano*)
  - 3.6.3. Definición y configuración de ambientes
  - 3.6.4. Ciclo típico de desarrollo y despliegue
- 4. Conclusiones
- 5. Bibliografía

## 5. Bibliografía propuesta

- [1] BECK, K. *Test-driven development*, 1 ed. Addison-Wesley, 2003.
- [2] BLACK, D. A. *The well-grounded Rubyist*, 1 ed. Manning, 2009.
- [3] BURNS, A., AND COPELAND, T. *Deploying Rails*, 1 ed. Pragmatic Bookshelf, 2012.
- [4] FOWLER, M. *Patterns of enterprise application architecture*, 1 ed. Addison-Wesley, 2003.
- [5] FOWLER, M., AND BECK, K. *Refactoring*, 1 ed. Addison-Wesley, 1999.
- [6] HENDERSON, C. *Building scalable web sites*, 1 ed. O'Reilly, 2006.
- [7] LEONARD, R. *RESTful Web services*, 1 ed. O'Reilly, 2007.

- [8] METZ, S. *Practical object-oriented design in Ruby*, 1 ed. Addison-Wesley, 2012.
- [9] OLSEN, R. *Design patterns in Ruby*, 1 ed. Addison-Wesley, 2008.
- [10] PERROTTA, P. *Metaprogramming Ruby*, 1 ed. Pragmatic Bookshelf, 2010.
- [11] RAPPIN, N. *Rails test prescriptions*. Pragmatic Bookshelf, 2010.
- [12] SOMMERVILLE, I. *Software engineering*, 9 ed. Pearson, 2011.
- [13] ZAKAS, N. C. *High performance JavaScript*, 1 ed. O'Reilly/Yahoo! Press, 2010.

---

Héctor Enrique Gómez Morales

---

M. en I. y M. en A.O. Karla Ramírez Pulido