Departamento de Computación y T.I

Ingeniería de Software I

Abril Julio 2013

Informe – Tarea 5 -

Django: modulo administracion y facturacion de servisoft

**Integrantes:**

Gamar Azuaje

Rosangelis García

José Luis Jiménez

Reinaldo Verdugo

Rebeca Machado

Gustavo El Khoury

Leopoldo Pimentel

Tabla de Contenidos

[1 Introduccion 3](#_Toc369803080)

[2 Diagrama de Clases 4](#_Toc369803081)

[3 Diagrama ER-E 5](#_Toc369803082)

[4 Diccionario de Datos 6](#_Toc369803083)

[5 Gestión de Riesgos 7](#_Toc369803084)

[5.1 Tecnología 7](#_Toc369803085)

[5.2 Personal 7](#_Toc369803086)

[5.3 Organizacional 7](#_Toc369803087)

[5.4 Herramientas 7](#_Toc369803088)

[5.5 Requerimientos 8](#_Toc369803089)

[5.6 Estimación 8](#_Toc369803090)

[5.7 Clasificación de los Riesgos 8](#_Toc369803091)

[6 Preguntas 9](#_Toc369803092)

[6.1 Pregunta #1 9](#_Toc369803093)

[6.1.1 ¿Qué recomendarían a la gerencia respecto al desarrollo de ServiSoft? 9](#_Toc369803094)

[6.2 Pregunta #2 9](#_Toc369803095)

[6.2.1 ¿Los patrones probados proporcionan alguna ventaja o beneficio interesante? 9](#_Toc369803096)

[6.3 Pregunta #3 9](#_Toc369803097)

[6.3.1 ¿Cuál sería el próximo paso? 9](#_Toc369803098)

[7 Conclusiones 10](#_Toc369803099)

[8 Bibliografía 11](#_Toc369803100)

# Introduccion

Hoy en día existe una gran cantidad de herramientas a la hora de programar y construir una página web, desde herramientas para la escritura del código; como lo son los IDE[[1]](#footnote-2) (Netbeans, Eclipse, Dreamweaver), también existen frameworks (Django, Rails, Cake) de desarrollo rápido de aplicaciones web. Estos frameworks vienen con funciones ya implementadas que ayudan al desarrollo de una aplicación, usualmente estas funciones tienen que ver con el manejo de la base de datos y control de acceso. También permiten la abstracción del código usando el paradigma de desarrollo: modelo, vista, controlador. Por último existen los CMS[[2]](#footnote-3)(PHP-Nuke, Wordpress). Estos Sistemas manejadores de contenidos son aplicaciones ya implementadas que solo les hace falta unas modificaciones de diseño. Luego se hace el deploy a algún ambiente de producción, y ya tendríamos la aplicación funcional y personalizada. Estos CMS, usualmente ofrecen un framework interno que permite la personalización del código y la mejoría de la misma.

Para la realización de esta tarea se tuvo que implementar una interfaz web que permitiera trabajar con la implementación ya realizada. Al mismo tiempo se necesitó la creación de módulos que permitieran la administración y la revisión de la factura por parte del usuario. Los administradores serian capaces de realizar cambios sobre los datos que maneja el sistema, mientras que los usuarios solo podrían visualizar la factura que representa su consumo mensual, dependiendo si esta es postpago y prepago.

Para facilitar la ejecución de esta tarea, se contó el apoyo del framework de desarrollo Django. Django está escrito en Python y actualmente se centra sobre el paradigma de desarrollo modelo, vista, controlador. [[3]](#footnote-4)

Django, adicionalmente contiene una gran cantidad de funciones dentro de su librería que permiten una implementación más rápida de los requerimientos antes mencionados. Como se mantiene el mismo lenguaje de programación en el que se hicieron las tareas 3 y 4, la implementación en este framework fue bastante sencilla en la parte de las clases y de los procedimientos.

# Diagrama de Clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, orientados a objetos.

# Diagrama ER-E

“Un diagrama o modelo entidad-relación (a veces denominado por sus siglas en inglés, E-R "Entity relationship", o del español DER "Diagrama de Entidad Relación") es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.”[[4]](#footnote-5)

# Diccionario de Datos

“Un diccionario de datos es un conjunto de metadatos que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.”[[5]](#footnote-6)

# Gestión de Riesgos

Cuando hablamos de la gestión de riesgos, primero tenemos que hablar de riesgos. Según el profesor Teruel, un riesgo “Es un evento o situación posible adverso al desarrollo de un proyecto. Es importante distinguir la *probabilidad* del riesgo y el grado de su posible *impacto” [[6]](#footnote-7)*

Cuando se analizaron los riesgos en nuestro equipo, pudimos identificar varios factores de riesgos, estos, los clasificamos dependiendo de su categoría.

## Tecnología

* Riesgos con el servidor de Django, (Servidor Web). Ya que aunque este es solo usado para hacer las pruebas, es algo nuevo para nosotros.
* Riesgos con el servidor de Postgres y la base de datos. (Servidor de Base de Datos).Aunque hemos usado Postgress para las ultimas tareas, para trabajar con Django existe un cambio en la forma que trabajamos con el servidor y por consiguiente es un riesgo a ser tomado en cuenta.
* Riesgos con Linux, se exigió un cambio de sistema operativo para trabajar, Varios miembros del equipo tuvieron que migrar a Linux, representando un riesgo de tecnología al no adaptarse lo suficientemente rápido.

## Personal

* Como riesgo personal, tomamos en cuenta que este proyecto salió las últimas semanas del trimestre. Estas últimas semanas todos estamos bastantes cortos de tiempo para aprender a usar nuevas tecnologías y al mismo tiempo para dedicarle a una materia en específico.
* Parte de ese riesgo, es que alguno del equipo no realice su parte del trabajo, dejando la entrega inconclusa

## Organizacional

* Para esta tarea, dejamos de trabajar en las parejas establecidas desde un comienzo del trimestre y nos repartimos el trabajo entre todos por igual. Cada uno eligió lo que quería hacer. Esto, representa un riesgo organizacional, ya que hay un cambio de responsabilidades y esto influye en el liderazgo que unos miembros del grupo tienen sobre otros.

## Herramientas

* Para esta entrega estuvimos trabajando con una nueva herramienta Django. En el equipo nadie había trabajado con esa herramienta previamente y existía el riesgo de no terminar la tarea por no poder finalizar los requerimientos para esta tarea.

## Requerimientos

* Para la realización de esta asignación, se nos pidieron varios requerimientos con respecto a la visualización. Se necesitan varias vistas entre las cuales, la parte de administración y la parte de usuarios. Como este requerimiento nos planteaba algo nuevo que no habíamos visto antes, representó un riesgo importante a la hora de hacer la gestión de riesgos.

## Estimación

* Riesgo de no terminar a tiempo la tarea
* Riesgo de infravalorar los cambios que se tienen que hacer en el sistema para que funcione todo correctamente.

Cada uno de estos riesgos, fue clasificado según su Probabilidad de Ocurrencia y su Impacto en caso de de ocurrir.

## Clasificación de los Riesgos

|  |  |  |
| --- | --- | --- |
| Riesgo | Probabilidad | Impacto |
| Riesgos con el servidor de Django | Baja | Catastrófico |
| Riesgos con el servidor de Postgres | Baja | Grave |
| Riesgos con Linux | Baja | Grave |
| Ultimas Semanas | Alta | Grave |
| Partes Inconclusas | Alta | Catastrófico |
| Repartición de las tareas | Media | Grave |
| Django | Alta | Grave |
| Requerimientos de Vistas | Media | Media |
| Tiempo para completar la tarea | Alta | Catastrófico |
| Infravalorar los cambios al sistema | Media | Grave |

# Preguntas

## Pregunta #1

### ¿Qué recomendarían a la gerencia respecto al desarrollo de ServiSoft?

* Para el correcto desarrollo de la aplicación, se recomienda seguir con la implementación actual. Y Dejar la base de datos como se encuentra, con los triggers. De esta forma existe información confiable en la base de datos.
* Tan bien recomendamos la ejecución de más pruebas, de forma de blindar la aplicación antes de salir a un ambiente de producción
* Crear una interfaz más agradable y mejorar la presentación grafica de la interfaz.
* Crear un modulo de administración completamente independiente del modulo de administración de Django.

## Pregunta #2

### ¿Los patrones probados proporcionan alguna ventaja o beneficio interesante?

* Si, los patrones probados ofrecen una ventaja a la hora de mejorar la cohesión y el acoplamiento del código.
* Con el patrón estrategia se pudo crear una mejor forma de facturar los clientes que tienen cuentas prepagadas y cuentas postpagadas. Con este patrón, ahora existe la facilidad de poder hacer algún cambio en alguno de los algoritmos que ejecuta esa funcionalidad independientemente.

## Pregunta #3

### ¿Cuál sería el próximo paso?

* El próximo paso sería terminar la aplicación. Completar los módulos que hacen falta.
* Crear un buen front end para los usuarios
* Crear un buen back end para los administradores
* Hacer una interfaz agradable a la vista con buenos diseños
* Hacer una prueba antes de sacar la aplicación al mercado.

# Conclusiones

Con la realización de este trabajo pudimos observar el funcionamiento de un nuevo framework, el cual se presta para la realización de los modelos agiles de desarrollo

# Bibliografía

**Diccionario de Datos** [En línea] / aut. Wikipedia Fundation Diccionario de Datos // Wikipedia. - http://es.wikipedia.org/wiki/Diccionario\_de\_datos.

**Django** [En línea] / aut. Wikipedia Fundation Django // Wikipedia. - http://es.wikipedia.org/wiki/Django\_(framework).

**IDE** [En línea] / aut. Wikipedia Fundation IDE // Wikipedia. - http://es.wikipedia.org/wiki/Entorno\_de\_desarrollo\_integrado.

**Laminas de Gestion de Proyectos** / aut. Teruel Alejandro. - Caracas : [s.n.], 2013.

**Modelo Entidad Relacion** [En línea] / aut. Wikipedia Fundation ER-E // Wikipedia. - http://es.wikipedia.org/wiki/Modelo\_entidad-relaci%C3%B3n.

**Sistema de Manejo de Contenidos** [En línea] / aut. Wikipedia Fundation CMS // Wikipedia. - http://es.wikipedia.org/wiki/Cms.

1. [↑](#footnote-ref-2)
2. [↑](#footnote-ref-3)
3. (Wikipedia Fundation Django) [↑](#footnote-ref-4)
4. (Wikipedia Fundation ER-E) [↑](#footnote-ref-5)
5. (Wikipedia Fundation Diccionario de Datos) [↑](#footnote-ref-6)
6. [↑](#footnote-ref-7)