

|  |
| --- |
|  |

**Informe del Proyecto**

**Planificación Semestral de Asignaturas UTEM**

|  |
| --- |
| **Integrantes** |
| **Carlos Guerrero Urbina** |
| **Daniel Gutierrez Pizarro** |
| **Ricardo López Núñez** |

|  |  |
| --- | --- |
| **Asignatura** | **Ingeniería de Software** |
| **Profesor** | **Sebastián Salazar Molina** |
| **Fecha** | **Lunes 30 de Junio,2014** |

**Tabla de Contenidos**

Tabla de Contenidos…………………………………………………………………………………………….........2

Introducción……………………………………………………………………………………………..........................3

1.1 Objetivos Generales……………………………………………………………………………………….……4

1.2 Objetivos Específicos…………….…………………………………………………………….………….……4

1.3 Alcances……………………………………………………………………………………………...........................5

1.4 Limitaciones……………………………………………………………………………………………..................5

1.5 Metodología……………………………………………………………………………………………..................6

1.6 Diseño de la Solución…………………………………………………………………………………………..9

1.6.1 Recursos Tecnológicos…………………………………………………………………………..10

1.7 Desarrollo del Proyecto……………………………………………………………………………………….11

1.8 Conclusiones……………………………………………………………………………………………..............

1.9. Bibliografía…………………………………………………………...………………………………………..…

2.0 Anexos………………………………………………………………………………………………………….

|  |
| --- |
|  |

**INTRODUCCIÓN**

En los últimos años hemos sido espectadores de un crecimiento exponencial de la demanda de software, que se ha venido aplicando en la resolución de tareas cada vez más complejas y proporcionando un mayor valor añadido. Los productos software siguen entregándose fuera de tiempo, excedidos en costo y no cumplen con la calidad y los objetivos esperados por el cliente. En estas circunstancias, se debe hacer la siguiente pregunta: ¿La industria del software está preparada para entregar aquel producto con los niveles de productividad y calidad que se requieran?

La educación sobre los futuros Ingenieros de Software, que ejercerán su actividad profesional en el ámbito público y/o privado son fundamentales para garantizar el crecimiento de la productividad de cualquier organización. Los alumnos del área de la informática y la computación siempre suelen recurrir a la Web para buscar los contenidos de sus asignaturas de la carrera, ya sean guías, controles, material teórico. Es aquí donde nos preguntamos si el Profesor en la UTEM ¿Está realizando correctamente su trabajo al ingresar una planificación semestral para que el alumno esté plenamente informado de lo que acontecerá durante el semestre en cuanto a los contenidos, evaluaciones y objetivos específicos de cada asignatura?.

Actualmente en la universidad la Vicerrectoría Académica junto con la DEA (Dirección de Evaluación Académica) crearon un documento exclusivo para la Planificación Semestral, lo que nos interesa principalmente es que si los, contenidos de las asignaturas presentadas por el Profesor en cuestión se cumplen a cabalidad durante todo el semestre, desde su inicio hacia el término.

**1.1 OBJETIVOS GENERALES**

El proyecto de planiﬁcación de clases, tiene como ﬁn llevar un registro de los contenidos que se impartirán clase a clase en cada una de las asignaturas además de también contener estrategias de enseñanza , actividades y evaluaciones cuando corresponda . El diseño debe contemplar la realidad Universitaria total (se diseña pensando en la UTEM). Este sistema debe tener roles claramente deﬁnidos y separados con alcances distintos.

**1.2 OBJETIVOS ESPECÍFICOS**

1. El sistema debe ser capaz de Generar la planificación clases a clase, usando el formato de la DEA.
2. El sistema debe permitir las operaciones CRUD sobre cada una las operaciones del sistema.
3. Debe proveer mecanismos de búsqueda y filtros, para obtener la planificación que corresponde según diversos criterios.
4. Debe integrarse a un servicio REST para obtener los datos de la Universidad y la autenticación.
5. El sistema debe manejar los usuarios, roles y opciones asociados a esta información.
6. El sistema debe disponibilizar un servicio REST con las mismas opcione sutilizadas para la búsqueda.

**1.3 ALCANCES**

* El rol informativo que cumplirá el sistema para la estructura organizacional de la Universidad, tanto Decanos, Jefes de Carrera, Directores de Departamento , Profesores , y por sobretodo con los Estudiantes, ya que ellos están directamente relacionados con la planificación semestral de cada asignatura y su correspondiente evaluación o actividades que se estime conveniente.
* También ayudará a llevar un registro de contenidos organizado y con objetivos claramente definidos para la ideal enseñanza a los futuros profesionales de nuestra Universidad, ya que es fundamental tener una clase definida al momento de llegar al Aula y comenzar la clase de cada asignatura.
* Como proporcionará un menú de búsqueda se podrá acceder a un contenido específico de un profesor “X” en la Semana “Y”, para ver si realmente se pasaron los contenidos esperados en la fecha consultada o por el profesor buscado.
* Cada semestre permitirá una actualización de contenidos temáticos para cada asignatura, modificando así el nivel de lo que se impartirá y esperando que con las estrategias de enseñanza puedan definir un avance en la educación a nivel de Universidad.

**1.4 LIMITACIONES**

* La información de cada planificación puede ser errónea no se puede asegurar que todos los profesores llenen los campos de forma correcta, dado que pueden informar algo que es completamente falso y de paso ser criticado por algún Director de Escuela o hasta por los mismos estudiantes.
* También influye la disposición que se tenga para poder preparar y definir los contenidos mínimos que debe tener cada carrera, la responsabilidad la tienen principalmente los profesores o coordinadores de cada asignatura, son ellos los que definen al final que es lo que contendrá cada planificación, esto a su vez está sujeto a la revisión de los Jefes de Carrera y Directores de Departamento al revisar individualmente la planificación semestral de cada asignatura.
* Además el que se entregue una planificación organizada con su respectivo contenido y fechas importantes, no asegura que un curso de estudiantes puede rendir como el profesor espera, existen muchos factores detrás del rendimiento universitario, por lo que una planificación semestral vendría a colaborar con sólo un eslabón de una serie de elementos para un buen aprendizaje y un buen rendimiento universitario.

**1.5 METODOLOGÍA**

Una metodología es un marco de trabajo, una forma ordenada y metódica de poder desarrollar un proyecto o una tarea en específica dentro de un lapsus de tiempo.

Además de ser un conjunto de buenas prácticas que se enfocan en el orden y el método, para éste proyecto en específico utilizamos la Metodología Ágil Scrum.

**Metodología de Desarrollo: Scrum**

***1. Roles***

* Product owner: Universidad Tecnológica Metropolitana (UTEM)
* Scrum Master: Ricardo López Núñez
* Equipo de Desarrollo: Carlos Guerrero Urbina y Daniel Gutierrez Pizarro
* Roles Auxiliares
* Managers: Profesores, Administrador.
* StakeHolders: Vicerrector Académico, Decano, Director de Departamento, Jefes de Carrera

***2. Daily Scrum o Stand-up meeting***

Estas reuniones las realizamos con el fin de vislumbrar los avances diarios para el proyecto, ubicar falencias, destacar fortalezas y continuar con el trabajo.

* Las reuniones fueron aproximadamente a las 12:00 A.M
* Las reuniones no excedieron una duración de 15 minutos a 20 minutos
* La reuniones fueron realizadas físicamente en el edificio M1 específicamente en el 4to Piso de los laboratorios de informática y en la sala de la asignatura
* Se realizaron otro tipo de reuniones cibernéticas, este tipo de reuniones alternativas fué vía Grupo de Facebook entre los integrantes del grupo, en donde también se plantearon avances, en cuanto a lo realizado diariamente por cada uno.

Durante las reuniones,, como es sabido nos planteamos las siguientes preguntas:

* ¿Qué has hecho desde ayer?
* ¿Qué es lo que harás hasta la reunión de mañana?
* ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo?

***3. Reunión de Planificación del Sprint (Sprint Planning Meeting)***

Al inicio del ciclo Sprint (Duración: 7 días), la Reunión de Planificación del Sprint se lleva a cabo de la siguiente forma:

* Seleccionamos que trabajo haremos
* Preparamos, como grupo, el Sprint Backlog que detalla el tiempo que tomará hacer el trabajo.
* Identificamos y comunicamos cuánto del trabajo es probable que se realice durante el actual Sprint , las tareas que llevaremos a cabo durante éste.
* Consideramos 4 horas como máximo en cada Sprint, dado que cada integrante aparte de realizar este proyecto tiene otras responsabilidades, que si bien no entorpecen el desarrollo normal del trabajo, se consideran para una mayor organización de éste.

Al final del ciclo Sprint, dos reuniones se llevarán a cabo: la “Reunión de Revisión del Sprint” y “Reunión de Retrospectiva del Sprint”

***4. Reunión de Revisión del Sprint (Sprint Review Meeting)***

* En esta reunión evaluamos los avances del proyecto y ciertas tareas que no se cumplieron cuando se estipulan en las Reuniones de cada Sprint.

***5. Retrospectiva del Sprint (Sprint Retrospective)***

* Hicimos una autoevaluación de cada etapa del Sprint para identificar ventajas y desventajas del trabajo que hicimos.
* Esta reunión durará como máximo 1 hora.

También se adjunta la siguiente guía de trabajo, que se hizo durante el proyecto, el tiempo de cada actividad es medido en horas-hombre y es aproximado

|  |  |  |
| --- | --- | --- |
| Actividades Principales | Actividades Complementarias | Tiempo Aproximado |
| Entrevista Director Departamento | Obtener Formato DEA entre otras informaciones | 1 hora hombre |
| Aprender PHP | Seguir tutoriales, realizar ejemplos básicos | 5 horas-hombre |
| Aprender Laravel | Instalación, Configuración y Pruebas , seguir tutoriales, recabar información | 12 horas -hombre |
| Realizar Informes (2) |  | 5 horas- hombre |
| Base de Datos y Modelo Entidad Relación | Diseñar base de datos y Modelo entidad relación | 8 horas-hombre |
| Desarrollo Proyecto | Tutoriales de Laravel, Php,Css | 30 horas-hombre |
| Diseño PHP | Tutorial de CSS, Obtener información de Javascript | 10 horas-hombre |

**1.6 DISEÑO DE LA SOLUCIÓN**

Para el desarrollo del código se utilizó el Framework Laravel 4.2, el cual posee arquitectura MVC (modelo, vista, controlador).

La información es proporcionada por las bases de datos de Dirdoc, mediante un servicio REST que nos facilitó el profesor.

La implementación fue llevada a cabo mediante el IDE Netbeans 8.0 con soporte para PHP, el cual es el lenguaje de programación utilizado.

Con el fin de simplificar el diseño de la solución se adjuntan las siguientes imágenes , principales del portal, tanto la página de inicio y el login.

Fig. 1 Portal de Inicio

Fig. 2 Login de Usuario

**1.6.1 RECURSOS TECNOLÓGICOS**

Se adjunta la tabla con los recursos tecnológicos que poseen los integrantes del Grupo 01 “Planificación Clase a Clase”

|  |  |
| --- | --- |
| Recurso | Características |
| Notebook HP 425 | * Sistema(s) Operativo(s): Windows 7 Professional / Ubuntu Linux 13.10 * Procesador: AMD Athlon II P340 Dual Core 2.20 GHz * RAM: 4 GB DDR3 * Disco Duro: 500 GB |
| Netbook HP Mini 110-3700 | * Sistema(s) Operativo(s): Ubuntu Linux 13.10 * Procesador: Intel Atom N455 Dual Core 1.6 GHz * RAM: 2GB * Disco Duro: 320 GB |
| Notebook HP 1000-1416LA | * Sistema(s) Operativo(s): Kubuntu Linux 13.10 / Windows 8.1 Single * Procesador: Intel Core i3 2.3 GHz * RAM: 4GB * Disco Duro: 500 GB |

**1.7 Desarrollo del Proyecto**

Las primeras actividades para comenzar el desarrollo se tuvieron que instalar programas necesarios en cada uno de los equipos disponibles.

Los software instalados fueron:

1. Laravel 4.2
2. Netbeans 8.0
3. PostgreSQL 9.3.4
4. Apache 2.4.9
5. PHP 5.5.1.2
6. pgAdmin 3
7. phppgadmin 9.3.4

Nuestro proyecto , antes de comenzar a desarrollarlo, se instalo el Framework Laravel, para ello antes de usar laravel fué necesario instalar Apache, PHP y PostgreSQL, generalmente algunos tutoriales en internet recomiendan usar o tener instalado LAMP (Linux, Apache, Mysql y PHP) que sería otra alternativa.

Además se necesita tener instalado Composer y para eso lo primero que se instala es Curl, para ello se abre la terminal y se teclea el siguiente comando:

**Código :**

sudo apt-get install php5-curl  
sudo apt-get install curl

Y después se procede a instalar composer:

**Código :**

curl -sS https:*//getcomposer.org/installer | php*  
mv composer.phar /usr/local/bin/composer

Laravel, como lo indica su documentación, necesita MCrypt PHP Extension, entonces se proced a instalarlo:

**Código :**

sudo apt-get install php5-mcrypt

Ahora se probará que mcrypt está incluido en PHP. Se ejecuta el siguiente comando:

**Código :**

/usr/bin/env php --ri mcrypt

En el caso que mostrara un mensaje como el siguiente:

**Código :**

**Extension** 'mcrypt' not present

Quiere decir que no está incluido, entonces se hace lo siguiente; se busca el archivo php.ini que debe estar en una ruta similar a:

*/etc/php5/cli/php.ini*

Y se procede a editar el archivo php.ini

**Código :**

sudo gedit php.ini

Se Agrega la siguiente línea, en el archivo php.ini:

*extension=mcrypt.so*

Y se reinicia Apache:

**Código :**

sudo service apache2 restart

Ahora se puede descargar e instalar Laravel. Primero se ubica en la raíz de donde se quiere que quede nuestro proyecto, por ejemplo:

**Código :**

cd /var/www

Y ejecutando el siguiente comando:

**Código :**

composer create-project laravel/laravel [**NombreProyecto**] --prefer-dist

Ahora se debe configurar Apache para que funcionen las URLs amigables, para eso vamos a buscar el archivo donde se tiene configurado nuestro host:

**Código :**

cd /etc/apache2/sites-available/

Para efectos de la instalación el archivo de configuración default se llama: **000-default.conf**

Se procede a editarlo y se agrega las siguientes líneas :

**Código :**

<Directory /var/www/TuProyecto/>  
Options Indexes FollowSymLinks MultiViews  
AllowOverride All  
Order allow,deny  
allow from all  
</Directory>

Se ejecuta el siguiente comando:

**Código :**

sudo a2enmod rewrite

Se reinicia Apache una vez más:

**Código :**

sudo service apache2 restart

Luego se acceda desde nuestro navegador a la siguiente

URL: localhost/miProyecto/public,

Si sale un error similar a éste o una pantalla en blanco:

**Código Error:**

*file\_put\_contents(/Users/winas/workspace/media\_unified\_url/media\_unified\_url/app/storage/meta/services.json) [function.file-put-contents]: failed to open stream: Permission denied.*

Se ejecuta el siguiente comando dentro de la carpeta raíz del proyecto:

**Código :**

sudo chown -R www-data:www-data app/storage

Para verificar si todo salió bien se debe mostrar la clásica pantalla en el navegador específicamente en la URL: localhost/miProyecto/public, Laravel debería mostrarnos el “Hola Mundo”de la siguiente forma: “You have arrived.”

Con ello ya se puede comenzar a poder desarrollar nuestro proyecto para la asignatura.

**1.8 CONCLUSIONES**

**1.9 BIBLIOGRAFÍA**

**2.0 Anexos**

Nota 1.1 y 1.2 : Dentro de los requerimiento no-funcionales y de dominio se prefiere no ponerlos por un tema de no alargar tanto el infor, además los requerimientos funcionales se trasnformaron en objetivos generales y específicos