|  |  |
| --- | --- |
|  | Sede Concepción Talcahuano  JetCat – Primera entrega – Anexos |
|  |  |
|  |  |
|  | Miguel Montero  Cristián De la Rivera |
|  |  |
|  | |  | | --- | |  | | Concepción, septiembre de 2012 | |

# Anexos

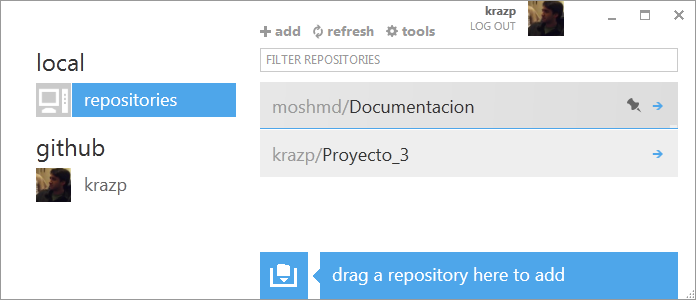
## Documentación de metodología usada

## Control de versiones a SW a entregar

El mecanismo principal de control de versiones para el desarrollo del proyecto está basado en el protocolo de git, para el cual se utiliza principalmente el software *github for Windows* y complementariamente la consola *Shell* de git (En casos específicos donde se requieran comandos especializados).

Básicamente, git puede funcionar de manera local o remota, donde todos los cambios realizados por cualquier miembro del equipo se van guardando de manera individual. Por otra parte si se esta trabajando con el mismo archivo en git, se comparan las líneas de del archivo (Como líneas de código cuando corresponde) y se guarda de forma individual. En el caso de que esto causara un conflicto se muestra un mensaje con una advertencia.

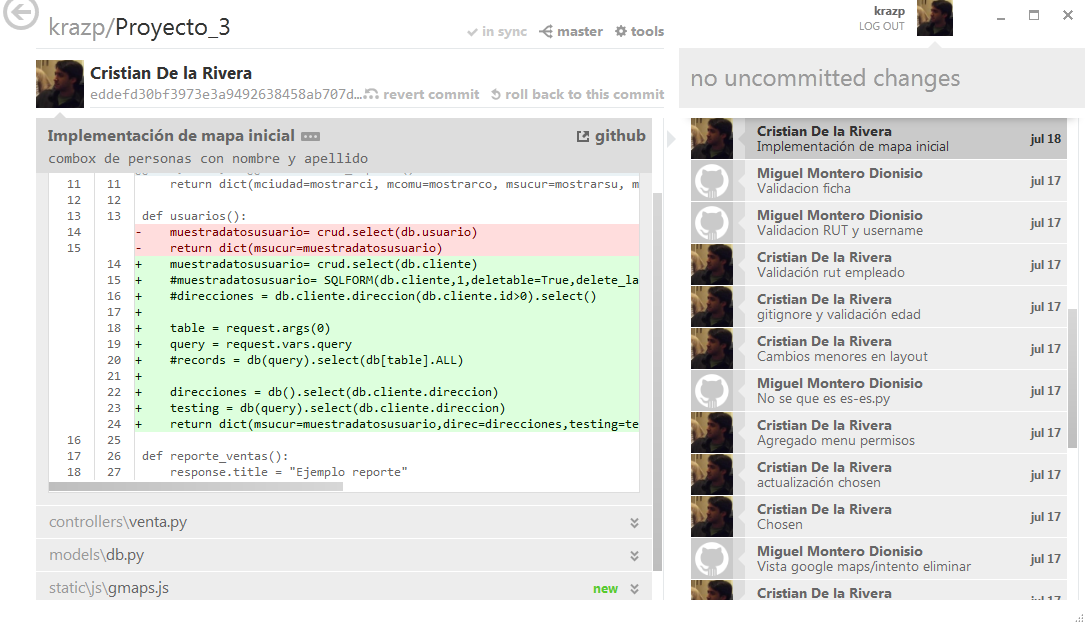
Capítulo 11, imagen 1



Menú principal de *github for Windows*

El control de software y documentación que se realiza a través de este medio permite la identificación de cambios de forma individual por cada miembro del equipo, además del detalle de los cambios realizado en cada *commit*. Además, existe la oportunidad de revertir el estado del elemento controlado a estados anteriores o revertir cambios específicos en caso de detectarse un error. En el desarrollo del proyecto, se maneja de forma separada el código fuente de la aplicación y la documentación.

Capítulo 11, imagen 2



Lista de cambios y detalle en *github for Windows*

Las posibilidades del software permiten analizar el progreso de cada miembro en particular y del proyecto en general, asistido por los nombres de cada ‘actualización’ o commit; en conjunto con las fechas. Una vez alcanzado hitos de programación avanzados, cada commit será identificado con un número de versión que se acerque a la versión 1.0 para la entrega del prototipo final, avanzando un número para cada adición de funcionalidades completas y decimales para la mejora de funcionalidades existentes.

## Documentación de seguimiento y controles

## Documentación de técnica de recolección de datos

## Planificación de control de riesgos identificados

**Problemas con entrevistas**

• Antes

* Se establece la reunión con cliente
* Revisión de fecha de reunión.

• Durante

* Formulario con datos de reunión
* Registrar problemas que ocurren dentro de la reunión dentro de una pauta.

• Después

* Hay que considerar que esta nueva entrevista debe ocurrir lo más pronto posible. Una vez acordado, se debe notificar al SQA, incluyendo la fecha de la siguiente reunión.

**Fallo de hardware o software**

• Antes

* Establecer plan de emergencia ante la contingencia, ya sea de HW o SW.
* Revisión de HW o SW.

• Durante

* Formulario de fallas de HW o SW.
* Registrar problemas en formulario de fallas de HW o SW.

• Después

* Evaluar datos registrados de fallos de HW/SW y tomar acciones correctivas correspondientes.
* Notificar al Jefe de Proyecto.

**Falta de hardware o herramienta de software**

• Antes

* Establecer el hardware o herramienta de software necesario.
* Revisión de hardware y herramienta de software.

• Durante

* Formulario de hardware y herramienta de software existentes.
* Registras inexistencia del HW o herramienta de software.

• Después

* Evaluar datos registrados de inexistencias y aplicar acciones correctivas.

**Problemas base de datos**

• Antes

* Se establece el tipo de base de datos a implementar, contemplando su robustez y funcionalidad
* Revisión de la base de datos desarrollada

• Durante

* Se cierran las conexiones de la base de datos y se identifica el problema en caso de error.

• Después

* Evaluar base de datos y aplicar acciones correctivas en caso de errores.

**Falta de equipo necesario**

• Antes

* Establecer el equipo necesario para el desarrollo del proyecto.
* Establecer pruebas parciales de los elementos involucrados en la implementación.
* Detectar posibles falencias o necesidades adicionales.
* Revisión de equipo, pruebas, falencias y/o necesidades adicionales.

• Durante

* Valida y verifica la existencia del equipo necesario.

• Después

* Se busca las opciones que existen de equipo que cumpla los requerimientos y se selecciona a opción que cumpla mejor las necesidades del proyecto. Esto se debe notificar al SQA si contempla un retraso significativo dentro del desarrollo del proyecto.

**Caída de servicios – Jefe de proyecto**

• Antes

* Se debe mantener organizado la información de los distintos servicios que utiliza el sistema.
* Revisión constante de los servicios que utiliza el sistema para su correcto funcionamiento.

• Durante

* Bajar el servicio que esta funcionando mal.
* Levantar el servicio después de solucionar la contingencia.
* Revisar operaciones anexas al servicio para verificar su correcto funcionamiento.

• Después

* Mantención de los servicios que se utilizan dentro del Software.

**Requerimientos no viables**

• Antes

* Establecer objetivos necesarios para el proyecto.
* Realizar estudio o análisis de técnica a utilizar para recolección de datos.
* Realizar entrevista para toma de requerimientos.
* Realizar investigación de factibilidades relacionadas al área de negocio del cliente.

• Durante

* Registrar problemas que ocurran en el análisis.

• Después

* En caso de que el conflicto sea detectado después, se debe contactar al cliente para informarle sobre conflicto, con posibles soluciones o alternativas disponibles que posee.

**Error en la planificación**

• Antes

* Establecer planificación temporal.
* Establecer tiempos de holgura para el desarrollo de la planificación.

• Durante

* Si se detecta una falla en la planificación, debe identificarse que ocasionó el problema y estimar el grado de daño que producirá esto sobre el desarrollo del proyecto.
* Registrar contingencias que ocurran dentro del desarrollo de la planificación.

• Después

* El jefe de proyecto toma medidas para neutralizar cualquier problema que pueda estar afectando la planificación y actualizar ésta para reflejar la nueva situación. Se debe incluir los cambios realizados en la planificación al cliente cuando esta ocasione cambios importantes.

**El sistema requiere más esfuerzo del presupuestado**

• Antes

* Establecer claramente las tareas y los tiempos.
* Evaluar el esfuerzo real requerido por tarea.
* Actualizar planificaciones.

• Durante

* Registrar problemas ocurridos dentro del tiempo en que se desarrolle el proyecto.

• Después

* Se deben buscar alternativas para aumentar la productividad del personal para mitigar los efectos de este problema sobre el desarrollo del proyecto.

**Cambio de requerimiento**

• Antes

* Establecer los requerimientos del usuario.
* Verificar que se cumplan los requerimientos dentro del proyecto.

• Durante

* Se debe analizar el requerimiento modificado por el cliente y evaluar el impacto que tiene sobre el diseño actual del software.

• Después

* Se debe identificar las partes del sistema comprometidas por el cambio y notificar al SQA. Este debe estimar el efecto que tendrá el cambio sobre el desarrollo y actualizar la planificación para reflejar estos cambios. Finalmente se rediseña las secciones involucradas para que cumplan con los nuevos requerimientos.

## Representación gráfica de planificación temporal