

# ¿Cómo clonar el repositorio?

1. Bajar netbeans la version full [aqui](#).

**NetBeans IDE 7.0.1 Download** 6.9.1 | 7.0.1 | [Development](#) | [Archive](#)

Email address (optional):

Subscribe to newsletters: ☒ Monthly ☐ Weekly  
☒ NetBeans can contact me at this address

IDE Language: English Platform: Windows  
Note: Grayed out technologies are not supported for this platform.

---

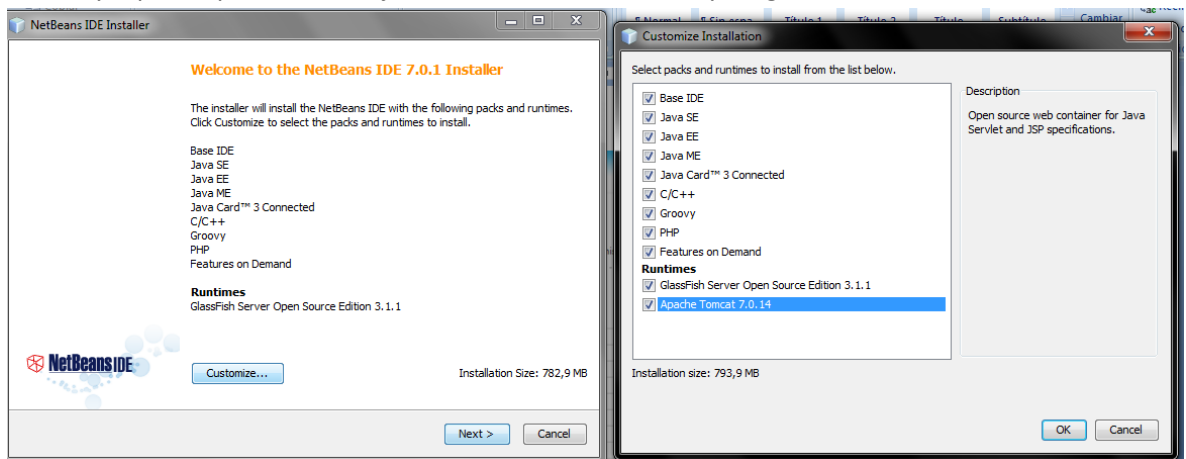
**NetBeans IDE Download Bundles**

Supported technologies *	Java SE	Java EE	C/C++	PHP	All
NetBeans Platform SDK	•	•			•
Java SE	•	•			•
Java EE		•			•
Java ME					•
Java Card™ 3 Connected					•
C/C++			•		•
Groovy					•
PHP				•	•
Bundled servers					
GlassFish Server Open Source Edition 3.1.1		•			•
Apache Tomcat 7.0.14		•			•
	<a href="#">Download</a> Free, 66 MB	<a href="#">Download</a> Free, 158 MB	<a href="#">Download</a> Free, 44 MB	<a href="#">Download</a> Free, 42 MB	<a href="#">Download</a> Free, 245 MB

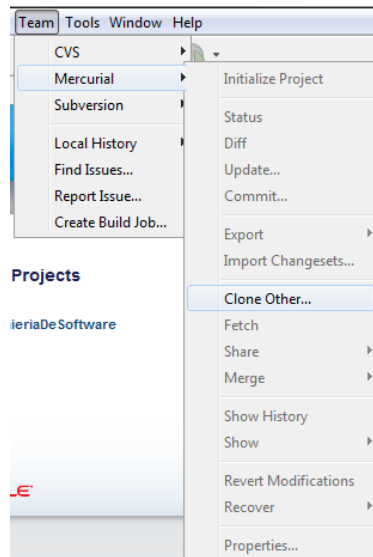
\* You can add or remove packs later using the IDE's Plugin Manager (Tools | Plugins). **Important Legal Information:**

2. Instalar Mercurial que se puede descargar [aqui](#).

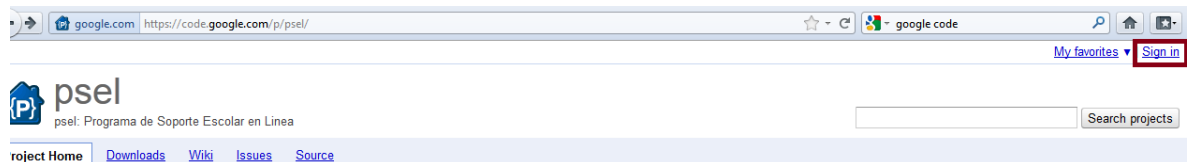
3. Cuando se vaya a iniciar el instalador asegurarse que se instale también **tomcat** este paso es vital ya que el que va a manejar los servlets será tomcat y no glassfish.



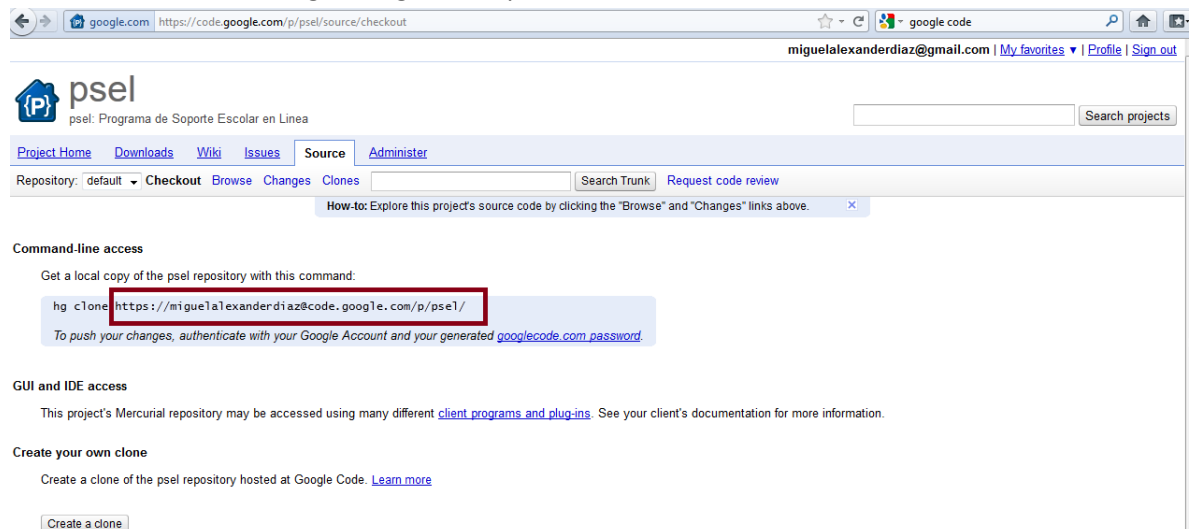
4. En netbeans acceder desde la barra de herramientas a Team->Mercurial->Clone Other...



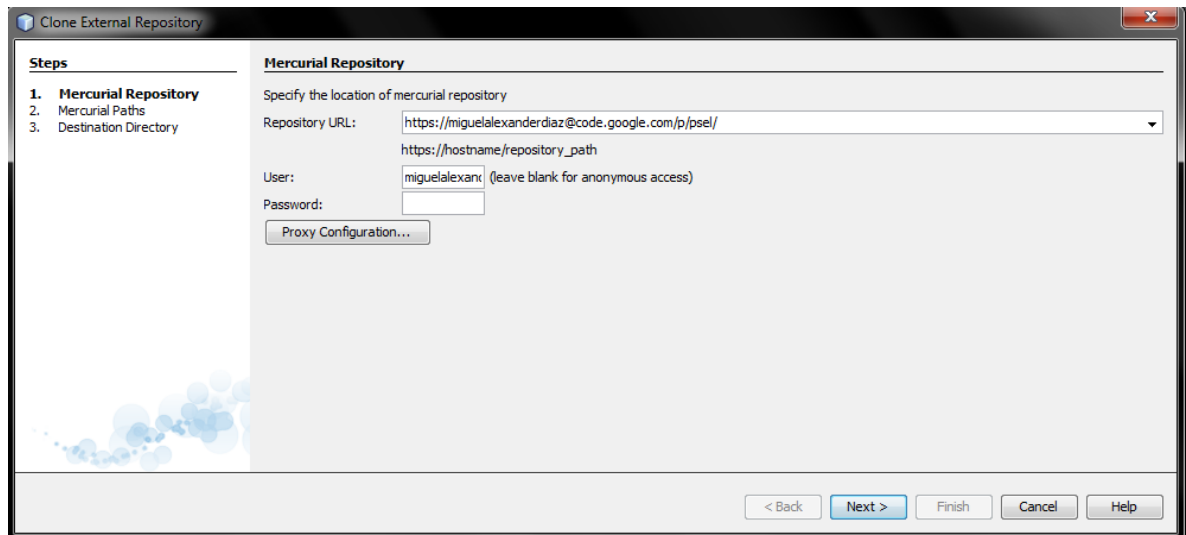
5. Ingresar al repositorio del proyecto en [googlecode](https://code.google.com/p/psel/) e ingresar con la cuenta de gmail de cada uno.



6. Una vez se ha hecho el log in dirigirse a la pestaña "source".



Allí encontraremos la dirección del repositorio para clonarlo en nuestras respectivas maquinas la dirección resaltada en la imagen se deberá pegar en el cuadro de dialogo del netbeans.



El user será el mismo de la cuenta de gmail en mi caso es [miguelalexanderdiaz@gmail.com](mailto:miguelalexanderdiaz@gmail.com)

El password **NO** es el mismo de la cuenta de gmail, esa información se encuentra en la página del repositorio y la encontraremos haciendo click en el link mostrado en la imagen de abajo.

#### Command-line access

Get a local copy of the psel repository with this command:

```
hg clone https://miguelalexanderdiaz@code.google.com/p/psel/
```

To push your changes, authenticate with your Google Account and your generated [googlecode.com password](#).

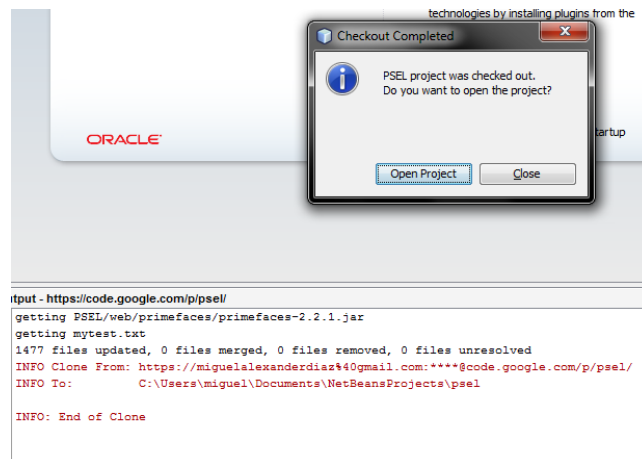
El password será el que nos muestre como en la imagen siguiente. Una vez llenados los campos del cuadro de dialogo de netbeans le damos next->next->finish.

#### GoogleCode.com Password

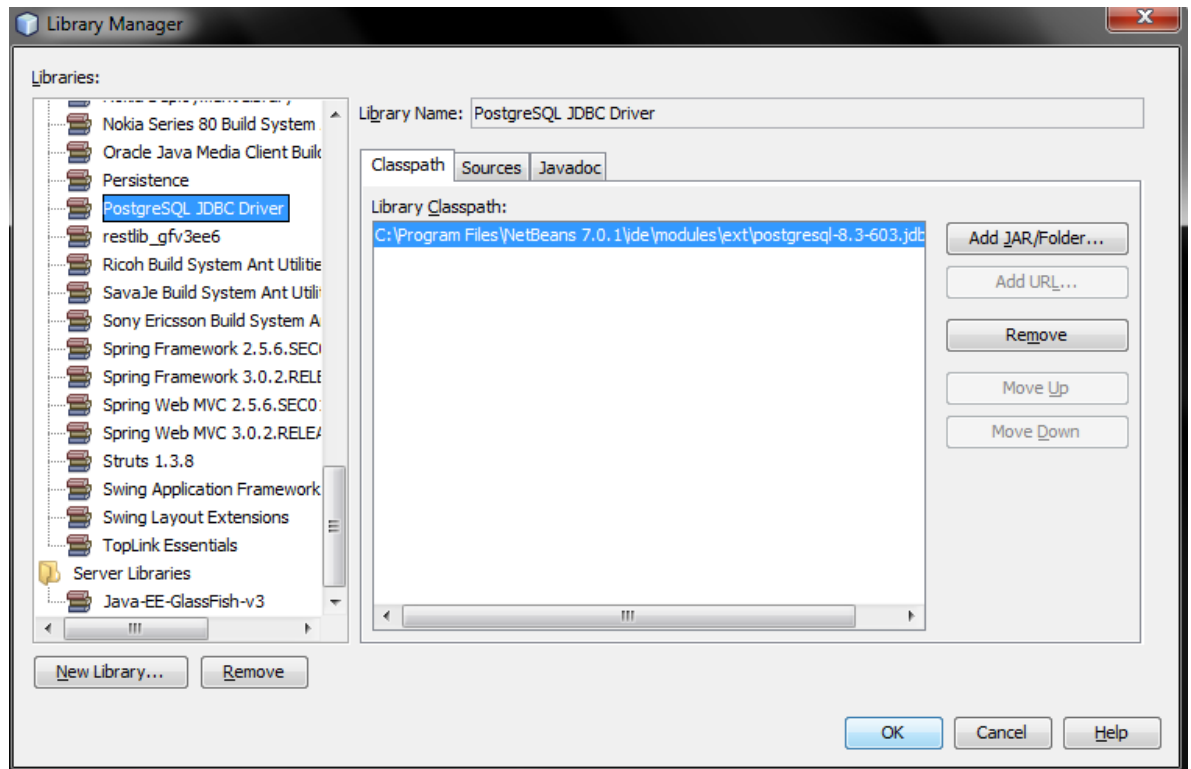
miguelalexanderdiaz@gmail.com's googlecode.com password: **gy3xs**

[Regenerate](#)

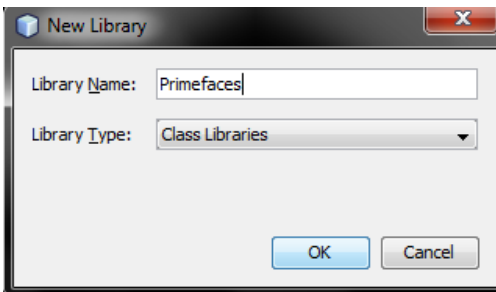
7. Una vez hecho esto obtendremos algo como lo siguiente.



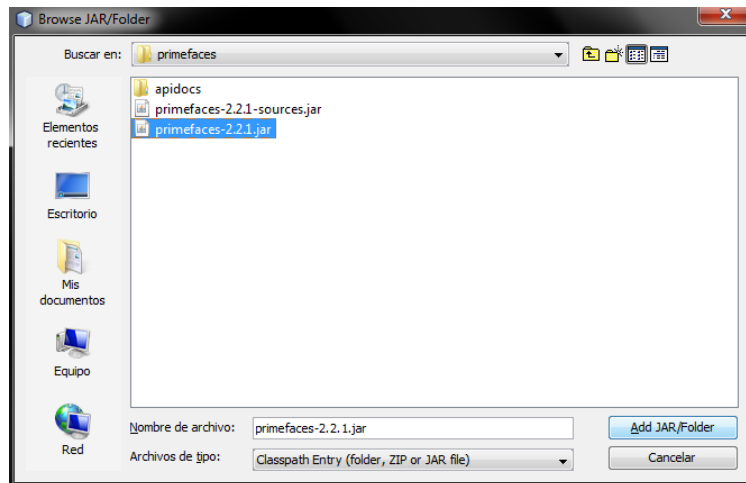
8. Cuando se abra el proyecto generara un error ya que el proyecto está utilizando una librería externa no estándar (Primefaces) la cual ha sido incluida en el repositorio para facilitar la configuración del proyecto. Para corregir este error le damos click en resolve y nos aparecerá algo como lo siguiente.



Le damos click en “New Library...” allí nos aparecerá un cuadro de dialogo y lo llenamos de la siguiente manera: como “Library Name:” le ponemos Primefaces y dejamos “Library Tipe:” Library. Y finalmente le damos OK.



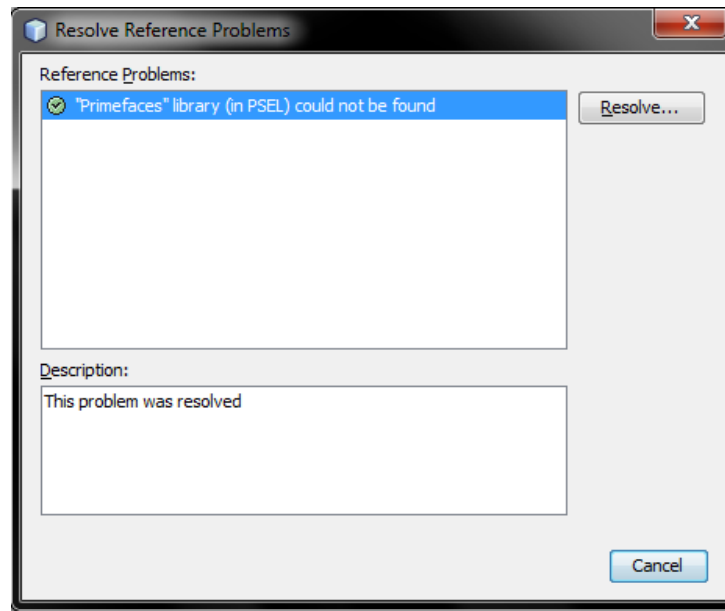
En la pestaña classpath le damos click al botón “Add JAR/Folder...” y vamos a la carpeta donde se guardo el proyecto y seleccionamos el archivo primefaces-2.2.1.jar, dicho archivo se localizara si se hizo todo como esta en el tutorial en:  
C:\Users\Nombre\_Usuario\Documents\NetBeansProjects\psel\PSEL\web\primefaces



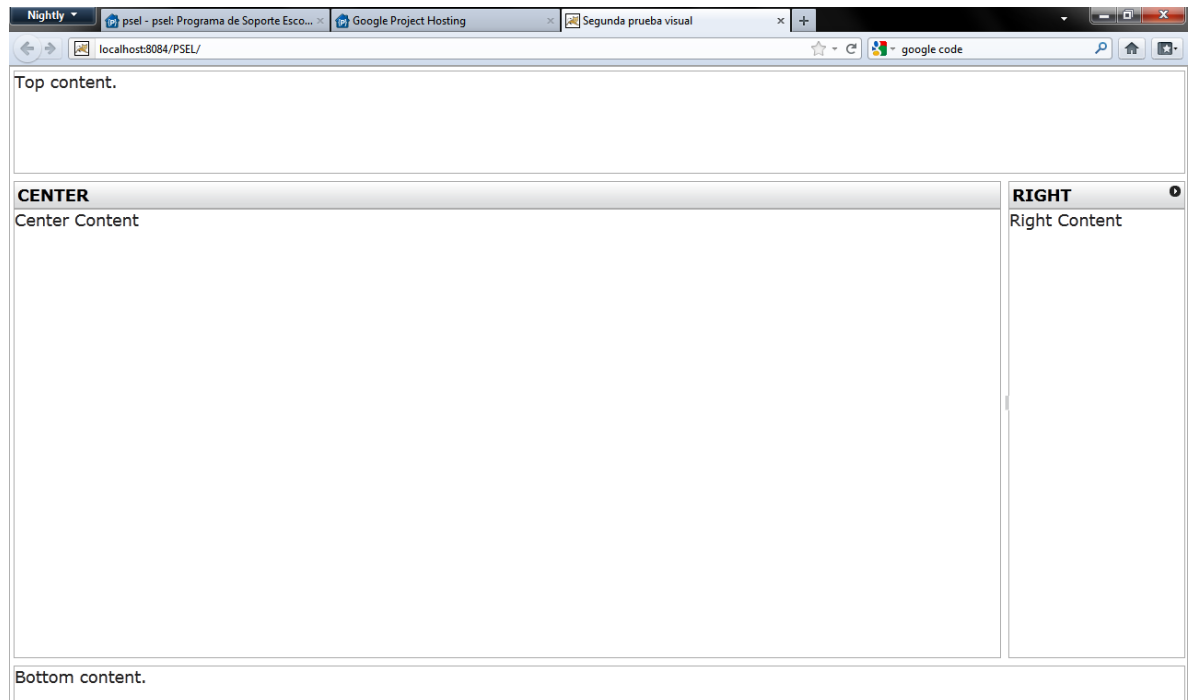
Luego vamos a la pestaña Sources y le damos click al botón “Add JAR/Folder...” y seleccionamos el archivo primefaces-2.2.1-sources.jar, el cual se localizara en la carpeta anteriormente señalada.

Luego vamos a la pestaña Javadoc y le damos click al botón “Add ZIP/Folder...” y seleccionamos la carpeta apidocs.

Finalmente le damos OK y obtendremos el siguiente mensaje.



Cerramos la ventana emergente y TENEMOS FINALMENTE EL PROYECTO FUNCIONANDO!  
Así ya podemos correr el proyecto para probarlo y obtendremos una página funcional que en la fecha de creación de este tutorial se ve algo así.



# ¿Cómo Trabajo con el repositorio?

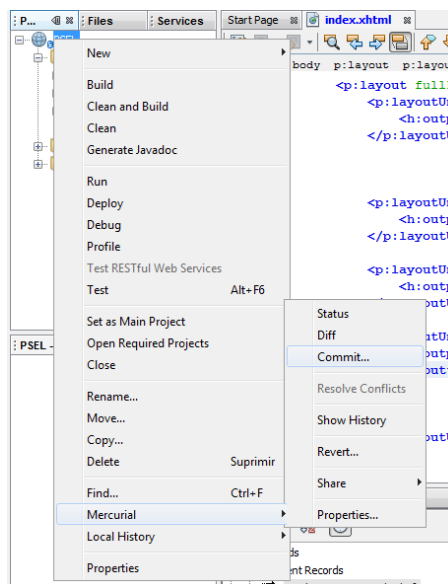
Para este tutorial debemos tener en cuenta primero las operaciones básicas de Mercurial.

- Commit: Establece los cambios en firme de forma estacionaria, es decir, que en el momento que damos commit el avance generado queda guardado en un chageset generando una entrada en el log del repositorio personal con el historial de cambios realizados de una versión a otra.
- Revert: Cuando hay una falla de una versión a otra es posible devolverse a un estado anterior y revertir los cambios que han ocasionado los fallos.
- Pull: Trae todos los datos nuevos que halla en el repositorio central (la nube) a nuestro repositorio personal.
- Push: Todos los cambios generados en nuestro repositorio personal se envían al repositorio central.
- Merge: Combina los datos de nuestro repositorio personal con los del repositorio central.

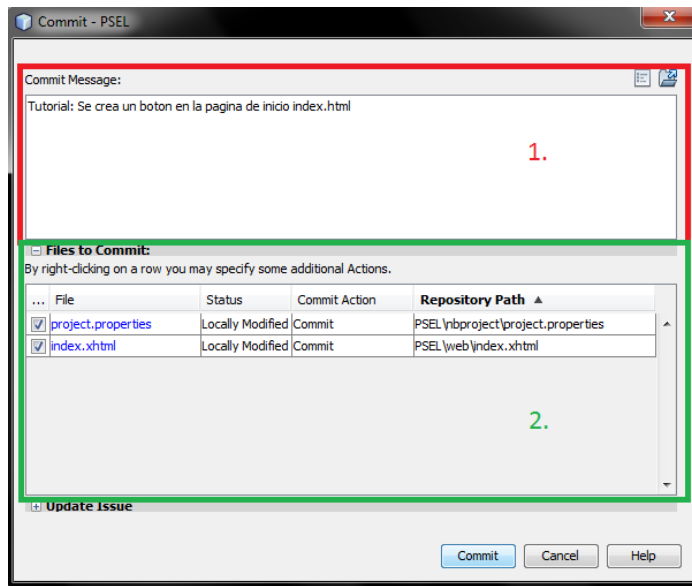
## *Generando un cambio en el repositorio personal:*

Una vez hecha una serie de cambios de medio alcance, como crear o modificar métodos, crear o modificar variables globales, modificar un aspecto de la GUI etc. Es necesario hacer un commit el cual siempre y sin excepción deberá contener una descripción detallada de que hay nuevo de una versión a otra, para ello deberemos seguir los siguientes pasos.

Supongamos que se crea un nuevo botón, y queremos guardar el cambio en firme en el repositorio personal, para ello le damos click secundario en el proyecto vamos a Mercurial->Commit...



Allí nos aparecerá el siguiente cuadro de dialogo:

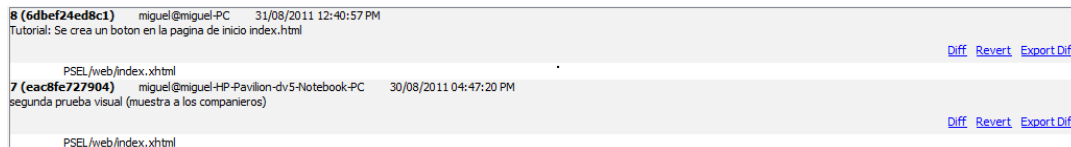


1. La parte marcada en rojo es sumamente importante, ya que debemos detallar lo mejor posible los cambios realizados en esta versión.
2. La parte en verde serán los archivos en los cuales se hizo una modificación efectiva, para este caso index.html donde se agrego el botón. Dichos cambios pueden ser visibles de una manera muy conveniente mediante Diff (descrito más adelante).

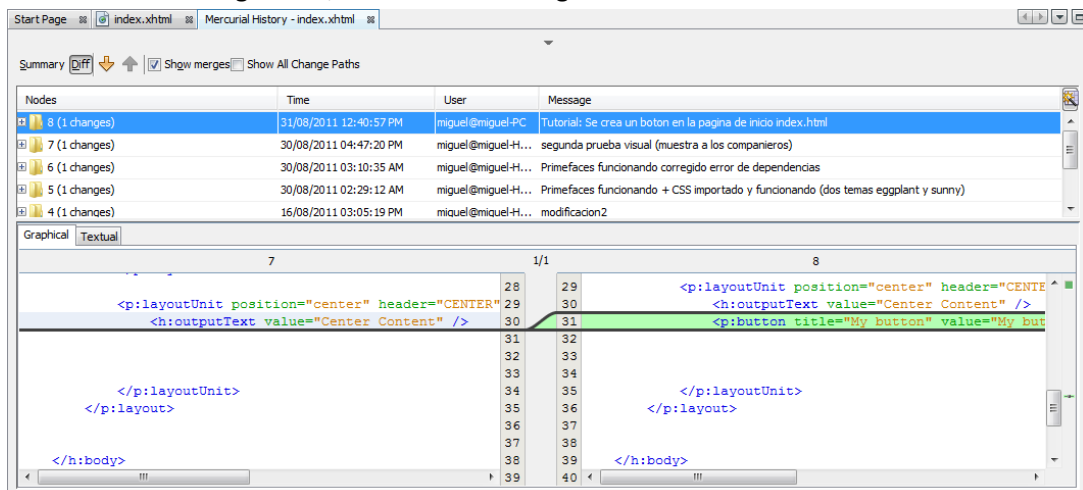
Una vez realizada la descripción y asegurarse de que todo es correcto le damos click al botón commit.

Podemos ver que efectivamente se ha creado una nueva entrada en:

Team->Mercurial->Show History donde veremos lo siguiente:



Si le damos click en Diff podremos ver las diferencias entre el changeset anterior y el que hemos acabado de generar, obtendremos la siguiente vista:



A la izquierda encontraremos la versión anterior y a la derecha la nueva; en verde aparecerán las adiciones, en rojo las eliminaciones, y en azul las modificaciones. Como vemos en la parte superior nos aparecen arboles para navegación en archivos cuando de



una versión a otra se modifican varios archivos podemos ver también las modificaciones que se hicieron archivo por archivo.

### *Generando un cambio en el repositorio central:*

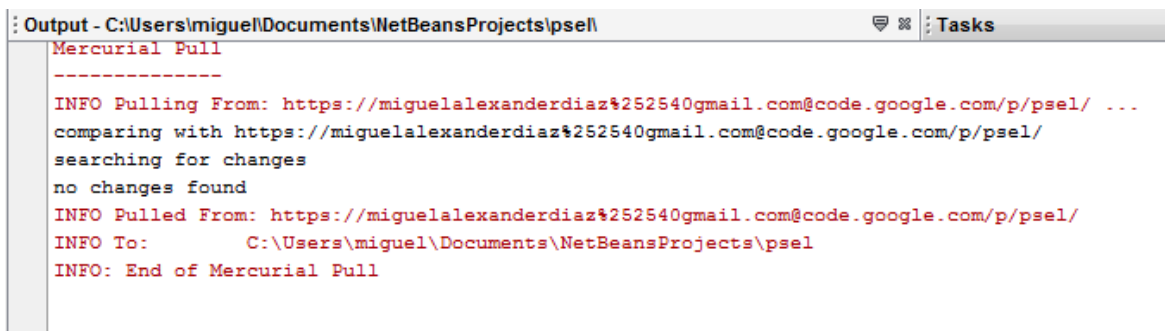
Cuando realizamos un cambio en nuestro repositorio personal este no actualiza de forma automática el central (en la nube) para ello debemos seguir la siguiente serie de pasos:

- Todas las contribuciones a subir al repositorio deben haber pasado primero por comando commit según el anterior numeral.
- Antes de subirlas se debe verificar que en el repositorio central no hayan nuevas modificaciones mediante comando pull si las hay generar un merge y corregir posibles conflictos.
- Una vez cumplidos los pasos anteriores mediante un comando push subir la contribución al repositorio central.

Para verificar que en el repositorio central no haya nuevas contribuciones se ejecuta un comando pull para ello damos click secundario en el proyecto,

Mercurial->Share->Pull from default-psel

de donde obtendremos la siguiente salida en la consola si no hay nuevas contribuciones.



```
Output - C:\Users\miguel\Documents\NetBeansProjects\psel\
Mercurial Pull
-----
INFO Pulling From: https://miguelalexanderdiaz%252540gmail.com@code.google.com/p/psel/ ...
comparing with https://miguelalexanderdiaz%252540gmail.com@code.google.com/p/psel/
searching for changes
no changes found
INFO Pulled From: https://miguelalexanderdiaz%252540gmail.com@code.google.com/p/psel/
INFO To: C:\Users\miguel\Documents\NetBeansProjects\psel
INFO: End of Mercurial Pull
```

Luego de esto debemos subir nuestra contribución a el repositorio central para ello ejecutamos un comando push, hacemos click secundario en el proyecto

Mercurial->Share->push to default-psel

Nos pedirá usuario y contraseña el usuario es el correo de gmail y la contraseña la suministrada por google code.

psel  
psel: Programa de Soporte Escolar en Linea

Project Home Downloads Wiki Issues Source Administrator

Repository: default Checkout Browse Changes Clones Search Trunk Request code review

Committed Changes					Today - Au
Rev	Scores	Commit log message	Date	Author	
c366587d7d34		Tutorial: Se crea un boton en la pagina de inicio index.xhtml	Today (13 minutes ago)	miguel@miguel-PC	

Si vamos a la página del repositorio podemos observar que el cambio en este caso la adición de un botón a la página se ha guardado en el repositorio central con éxito. En este caso se hizo la adición de un botón en la página obteniendo la siguiente vista.

Segunda prueba visual

localhost:8084/PSEL/

Facebook YouTube Google Wave SIA Vagos DirectorioWare Hacker tuto Python Lib. Prog Algoritmos 2011-1 Groovesark Grupo F! Otros marcadores

Top content.

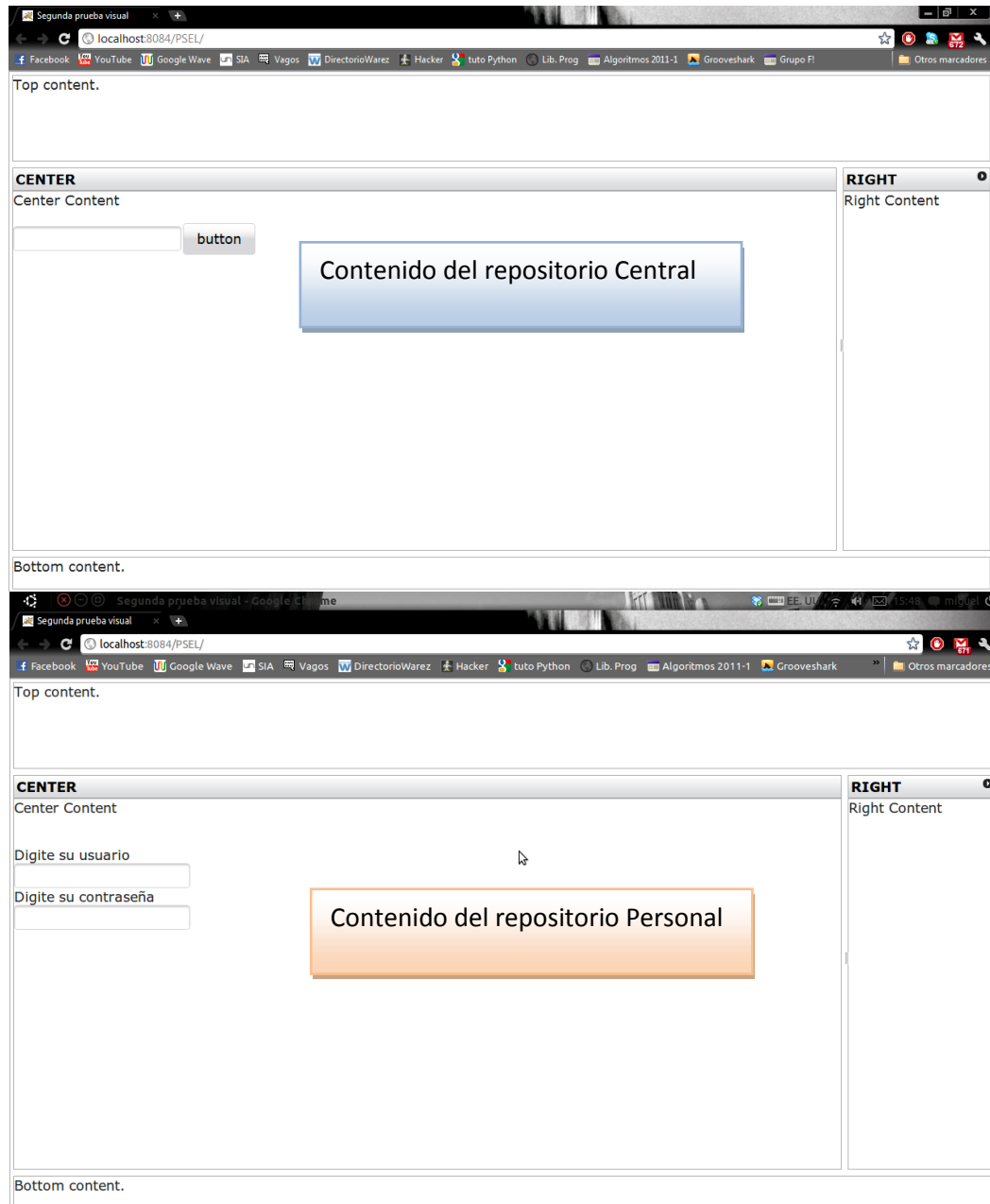
**CENTER**  
Center Content

**RIGHT**  
Right Content

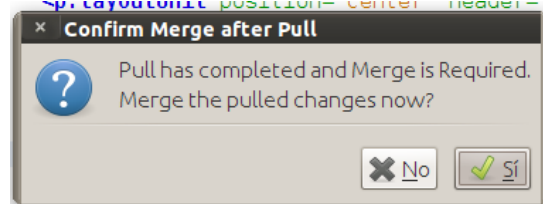
Bottom content.

### Uniando contribuciones:

Supongamos que he hecho un campo de texto y alguna descripción para el usuario en index.xhtml y no estoy al tanto de que el otro desarrollador hizo el botón del numeral anterior y lo ha subido al repositorio, tendremos una situación como la siguiente:

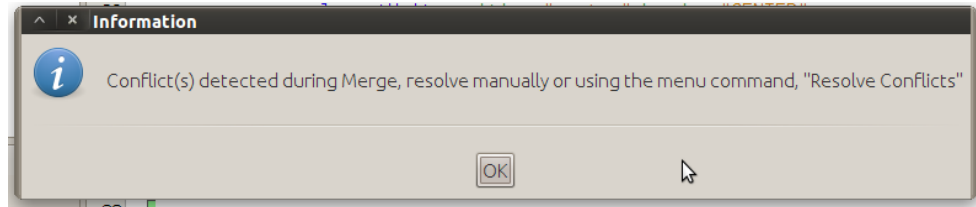


Ahora supongamos que quiero subir esta nueva contribución efectuando los pasos del anterior numeral, con la salvedad que ahora si encontraremos que hay un cambio. Una vez ejecutemos el comando pull nos saldrá el siguiente aviso.

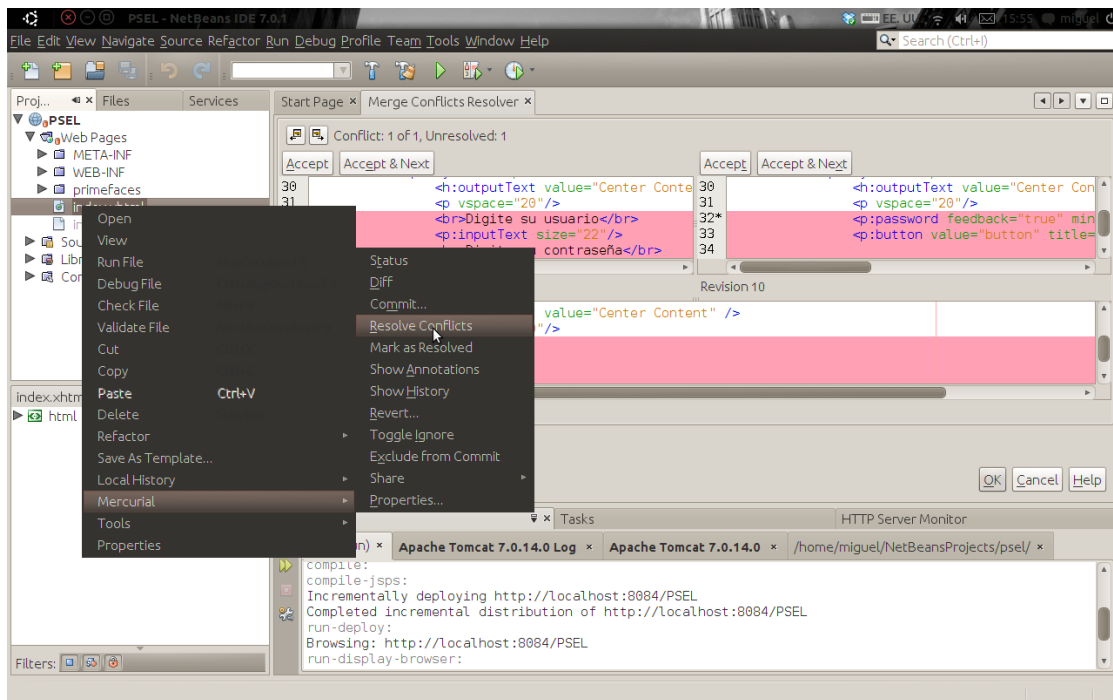


Le damos si, el proyecto ahora tendrá las contribuciones nuestras más las contribuciones ya establecidas en el repositorio, esto en el caso que no se presenten conflictos, de ser así generamos un commit colocando en la descripción la nota que se hizo un merge y posteriormente ejecutamos un comando push para subir el conjunto de cambios nuestros sumado a los cambios hechos por los demás.

Si se llega a presentar un conflicto nos aparecerá un mensaje como el siguiente que básicamente nos informa de los conflictos y que deben ser resueltos antes de continuar.



Para resolverlos le damos click secundario al proyecto Mercurial->Resolve Conflicts



Allí se desplegará una pantalla con 3 secciones la de la izquierda representa lo que existe en el repositorio personal y la de la derecha lo que existe en el repositorio central, y la inferior representa el código que está siendo fabricado en el momento de la resolución de los conflictos; para resolver dichos conflictos solo basta seleccionar una de las líneas de código ya sea de la izquierda o de la derecha dar click en Accept & Next hasta finalizar, una vez se hayan resuelto todos los conflictos le damos click en OK.

Ya resueltos todos los conflictos podemos subir los cambios con tranquilidad luego de hacer primero el respectivo commit.