

# **WIRA**

## **Manejo del Ambiente Controlado**

### **Versión 2.0.1**

#### **Semana 04**

#### **Grupo 11**

### **Historia de revisiones**

Fecha	Versión	Descripción	Autor
08/09/2013	1.0	Creación del Documento	Gonzalo Waszczuk
08/09/2013	1.0.1	Revisión SQA	Gonzalo Antúnez
12/09/2013	2.0	Cambio de servicio web, y otras modificaciones	Gonzalo Waszczuk
12/08/2013	2.0.1	Revisión SQA	Gonzalo Antúnez

# Contenido

<b>1.DEFINICIÓN DEL AMBIENTE CONTROLADO.....</b>	<b>3</b>
1.1.DEFINICIÓN DE REPOSITORIOS GIT	3
1.2.GOOGLE DRIVE	4
1.3.CLOUDFORGE	5
<b>2.DESCRIPCIÓN DEL USO DEL AMBIENTE CONTROLADO.....</b>	<b>5</b>
2.1.USO DE CLOUDFORGE	5
2.2.GIT	6
2.3.SETEAR GIT EN LINUX	6
2.4.SETEAR GIT EN WINDOWS 7 o XP	7
2.5.HERRAMIENTAS DE GIT	8
2.5.1.Git CLI (Command Line Interface):.....	8
2.5.2.Git Cola:.....	8
2.5.3.Source Tree:.....	9
2.5.4.Tortoise Git:.....	9
2.5.5.GitEye:.....	9
2.5.6.Integración git de Aptana Studio 3.....	10
2.6.CLONAR UN REPOSITORIO GIT	10
2.6.1.Git CLI.....	10
2.6.1.1.Git Cola.....	11
2.6.1.1.SourceTree.....	11
2.6.1.1.TortoiseGit.....	11
2.6.1.2.GitEye.....	11
2.6.1.3.Aptana Studio 3.....	12
<b>4.DESCRIPCIÓN DEL USO DEL AMBIENTE DE TRABAJO.....</b>	<b>12</b>
4.1.HERRAMIENTAS	12
4.1.1.Cacao.....	12
4.1.2.Gantter.....	12
4.2.SUGERENCIAS	13
<b>5.RESPONSABILIDADES.....</b>	<b>13</b>

## 1. Definición del Ambiente Controlado

El ambiente controlado es el entorno que se utiliza para almacenar los elementos del sistema, la línea base, el código fuente, y otros aspectos del sistema que sea necesario tener bajo control. A su vez se deben definir las herramientas y procedimientos para poder acceder a él y utilizarlo, y se deberá definir cómo utilizar tales herramientas.

Es necesario tener un ambiente de control simple de utilizar, y proporcionar las herramientas necesarias para hacerlo. De esta forma se simplifica el mantenimiento de cada elemento del sistema, y facilita la realización de las actividades de cada integrante del equipo.

A su vez, se define un ambiente de trabajo, fuertemente relacionado al ambiente controlado, para facilitar el trabajo de cada integrante del equipo al crear documentos, coordinar el trabajo sobre ellos, y poder realizar determinadas actividades del proyecto de forma sencilla.

Para la definición de estos ambientes, el Responsable de Gestión de Configuración (SCMR de ahora en adelante) cuenta con una cuenta administradora en cada uno de ellos. Tal cuenta será llamada "pis2013gr11", y cada cuenta estará asociada a la cuenta de Gmail "[pis2013gr11@gmail.com](mailto:pis2013gr11@gmail.com)". El único integrante que tendrá acceso a estas cuentas será el SCMR, pero si es necesario que algún integrante tenga acceso a alguna por algún motivo, se deberá comunicar con el SCMR y se podrá gestionar su uso por parte del integrante.

El ambiente de control está implementado utilizando varios repositorios git. Éstos están hospedados en CloudForge (<http://www.cloudforge.com/>). A su vez, parte del ambiente de control está implementado utilizando una carpeta compartida en Google Drive.

A continuación se define el uso de estos ambientes.

### 1.1. Definición de repositorios git

En el servicio de CloudHost se cuenta con 4 repositorios git:

1. Repositorio para documentación y elementos del sistema.
2. Repositorio para la línea base.
3. Repositorio para el desarrollo del producto.
4. Repositorio para la implementación del prototipo.

El repositorio de elementos del sistema será utilizado para mantener todos los elementos actuales del sistema y del proyecto. Esto incluye documentos, planillas, diagramas, etc. También se mantendrá un histórico de commits para el fácil acceso a cualquier versión de cualquier elemento del proyecto de forma fácil y sencilla.

El repositorio de la línea base será utilizado para guardar todos los elementos de la línea base en el momento actual del proyecto.

El repositorio para el desarrollo del producto será utilizado para que los implementadores puedan desarrollar el producto final. Este repositorio incluye el código fuente, y a su vez cualquier archivo necesario para que cada implementador pueda implementar de forma adecuada.

El repositorio del prototipo establece el ambiente de implementación de éste por parte de los especialistas técnicos. Incluye el código fuente y cualquier archivo necesario para poder desarrollarlo.

Por más información acerca de procedimientos, políticas de permisos, etc, sobre estos repositorios, ver el Plan de Configuración.

## 1.2. Google Drive

En Google Drive se cuenta con una carpeta compartida llamada "PIS", a la cual tienen acceso todos los integrantes del equipo. Esta carpeta pertenece a la cuenta de administración del SCMR de Gmail ([pis2013gr11@gmail.com](mailto:pis2013gr11@gmail.com)).

Esta carpeta cuenta con la siguiente estructura:

- Documentos
- Diagramas
- Plan de Iteración
- Registro de Horas

En la carpeta de *Documentos* se encuentra una estructura de carpetas que refleja cada disciplina/área del proceso MUM (e.g "Requerimientos"). En cada una de estas carpetas podrán existir archivos de Google Docs para poder trabajar sobre documentos de forma online, y para que varios integrantes del equipo puedan colaborar sobre ellos. También se podrán subir archivos auxiliares o necesarios para que los responsables de esas áreas puedan trabajar. Se sugiere que el nombre de los documentos a ser trabajados sea "[**Nomenclatura**] – [**Descripción**]", siendo [**Nomenclatura**] la nomenclatura del documento, y [**Descripción**] la descripción del mismo. E.g: "SCMMAC – Manejo del Ambiente Controlado". Se utilizará la herramienta Google Docs para trabajar sobre los documentos.

En la carpeta de *Diagramas* se encuentra la misma estructura de carpetas, que sigue las disciplinas/áreas del proceso MUM. Dentro de cada carpeta se deberán incluir los diagramas UML que están siendo creados y/o modificados en tales áreas del proyecto. Se utilizará la herramienta Cacoo para realizar tales diagramas y que distintos integrantes puedan colaborar sobre ellos.

En la carpeta *Plan de Iteración* se encuentra el plan de iteración de la iteración actual, para que pueda ser accedido y/o modificado por cualquier integrante del equipo. A su vez se encuentra una carpeta llamada "Otras Iteraciones", donde se encuentran los planes de iteración de iteraciones anteriores. El propósito de estas carpetas es para que en cualquier momento se tenga acceso a documentos anteriores del Plan de Iteración para poder editarlos y/o trabajar sobre ellos (ver la sección "Recuperación de Elementos del Sistema" del Plan de Configuración por más información). Se utilizará la herramienta Gantter for Google Drive para trabajar sobre el plan de iteración.

En la carpeta *Registro de Horas* se encuentra el registro de horas de la semana actual, para que pueda ser accedido por cualquier integrante del equipo. A su vez se encuentra una carpeta llamada "Otras Semanas", donde se encuentran los registros de horas de semanas anteriores. El propósito de estas carpetas es para que en cualquier momento se tenga acceso a registros de horas pasados para poder editarlos y/o trabajar sobre ellos (ver la sección "Recuperación de Elementos del Sistema" del Plan de Configuración por más información). Se utilizará la herramienta Google Spreadsheets para trabajar sobre el registro de horas.

La carpeta compartida de Google Drive a su vez establece el ambiente de trabajo del equipo. En este ambiente cualquier integrante puede definir sub-carpetas, archivos, documentos, etc, que crea necesarios para poder trabajar sobre alguna actividad del proyecto, o para llevar a cabo algún objetivo en particular. Se sugiere recibir permiso de la Administradora ([ceci714@gmail.com](mailto:ceci714@gmail.com)) y/o del SCMR ([gonzaw308@gmail.com](mailto:gonzaw308@gmail.com)) para crear o modificar la estructura del ambiente en Google Drive. En particular, ningún

integrante, sin ser la Administradora o el SCMR podrá eliminar o modificar los elementos de las carpetas "Registro de Horas" y "Plan de Iteración", ni modificar la estructura de archivos y carpetas descrita anteriormente.

### 1.3. CloudForge

Sitio: <http://www.cloudforge.com/>

CloudForge es un servicio web de hospedaje de proyectos. Estos proyectos incluyen repositorios git, herramientas integradas, y otras funcionalidades. Se utilizará este servicio para hospedar los repositorios git mencionados anteriormente, y para poder acceder a archivos de ellos por su interfaz web.

## 2. Descripción del uso del Ambiente Controlado

### 2.1. Uso de CloudForge

Cada integrante del grupo tendrá una cuenta en CloudForge. El SCMR es el encargado de crear las cuentas de todos los integrantes del equipo. A continuación se describen las cuentas de cada integrante del equipo:

Nombre	Apellido	Username	Password
Agustín	Azzinnari	aazzinnari	pis2013gr11
Alejandro	Cardone	acardone	pis2013gr11
Cecilia	Guayta	cguayta	pis2013gr11
Diego	Rey	drey	pis2013gr11
Germán	Wolman	gwolman	pis2013gr11
Gonzalo	Antunez	gantunez	pis2013gr11
Gonzalo	Waszczuk	gwaszczuk	pis2013gr11
Jimena	Pintos	jpintos	pis2013gr11
Juan	Bertoni	jbertoni	pis2013gr11
Matías	Carro	mcarro	pis2013gr11
Miguel	Merlino	mmerlino	pis2013gr11
Pablo	Scleidorovich	pscleidorovich	pis2013gr11
Sebastián	Sanchez	sebasanchez	pis2013gr11
Viterbo	García	vgarcia	pis2013gr11

Para poder loguearse, es necesario incluir un "domain". El "domain" del proyecto es **pis2013gr11**.

La primera vez que un integrante accede a su cuenta, éste deberá cambiar su contraseña lo antes posible. Para ello, una vez logueado deberá ir a la sección "My Settings", ubicada arriba de la página (bajo el ícono del usuario). Dentro deberá cambiar su contraseña.

En CloudForge, un usuario puede pertenecer a uno o varios proyectos. Los proyectos contienen un repositorio git, y formas de acceder a él. Cada proyecto tiene un dueño. En este caso, el dueño de cada proyecto es la cuenta administrativa "pis2013gr11" creada por el SCMR.

Para acceder a los proyectos de uno, se deberá acceder al "Dashboard" e ir a la sección "Projects" ubicada abajo de la página.

En la página principal del proyecto, se destaca la URL del repositorio a la derecha de la página (utilizada para poder clonar el repositorio). A su vez, debajo de la sección "Services" se encuentran los servicios del proyecto. Al

hacer click sobre el proyecto "Git", se puede acceder a la interfaz web del repositorio seleccionando "Browse with GitWeb". En la nueva pantalla se deberá ingresar el nombre de usuario y contraseña si es necesario, y luego se podrá recuperar cualquier archivo del repositorio accediendo al tab "tree" que se encuentra arriba de la página.

Si se necesita una versión antigua de un archivo, es posible acceder al histórico yendo al tab "commits" dentro del proyecto, y seleccionar el commit donde está tal archivo. También se pueden acceder a los archivos pertenecientes a un "tag", accediendo al tag apropiado en la sección "tags", y luego accediendo al tab "tree" que se encuentra arriba de la página.

## 2.2. Git

El ambiente de control está implementado utilizando repositorios git. Para poder utilizar el ambiente controlado, entonces es necesario saber cómo utilizar git para poder acceder a tales repositorios, trabajar sobre ellos, y actualizar tales repositorios.

Se recomienda leer las siguientes guías para entender los fundamentos de git, y cómo utilizarlo:

Guía oficial: <http://git-scm.com/documentation>

Guía visual sobre los fundamentos de git: <http://gitolite.com/gcs/index.html>

Otra guía visual: <http://marklodato.github.io/visual-git-guide/index-en.html>

Otra guía: <http://www.ralfebert.de/tutorials/git/>

## 2.3. Setear Git en Linux

Para poder setear git en Linux, y para poder setear las configuraciones necesarias para poder comunicarse con el servicio de CloudForge, se deberán seguir los siguientes pasos:

1. Descargar git. Para ello abrir una terminal y ejecutar el siguiente comando:  
**apt-get install git-core**
2. Revisar si se tiene OpenSSH instalado. Para ello ejecutar el siguiente comando en una terminal:  
**ssh -v.**  
Si la salida muestra una versión de OpenSSH instalada, entonces seguir al siguiente paso. De lo contrario instalar OpenSSH con el comando:  
**apt-get install openssh**
3. Crear claves de SSH. Abrir una terminal y ejecutar el comando:  
**ssh-keygen -t rsa.**  
Elegir todas las opciones por defecto cuando se lo pida (apretar Enter cada vez). Esta acción crea un par de claves ssh en la carpeta ~/.ssh . Estas claves tienen el nombre "id\_rsa" y "id\_rsa.pub" (para la clave privada y pública respectivamente). Si se pregunta para sobrescribir tales claves, entonces ya se cuenta con una clave ssh, por lo tanto se puede seguir al siguiente paso.
4. Copiar clave pública. Para ello ejecutar este comando, y copiar al portapapeles su salida:  
**cat ~/.ssh/id\_rsa.pub**
5. Setear claves en la cuenta de CloudForge. Para ello ir a la cuenta CloudForge, e ir a My Settings->Keys->Add SSH Key. Ingresar la clave que se tiene copiada en el portapapeles, e ingresar un nombre descriptivo para ella (se sugiere identificar tales claves si se va a acceder a los repositorios en distintas máquinas).

6. Setear nombre y email en git. Para ello ejecutar estos comandos en una terminal:

**git config --global user.name [nombre]**

**git config --global user.email [email]**

Donde **[nombre]** es el nombre del integrante, y **[email]** su email.

## 2.4. Setear Git en Windows 7 o XP

Para poder setear git en Windows 7 o XP, y para poder setear las configuraciones necesarias para poder comunicarse con el servicio de CloudHost, se deberán seguir los siguientes pasos:

1. Instalar msysgit:
  1. Descargarlo desde la siguiente url:  
<https://code.google.com/p/msysgit/downloads/detail?name=Git-1.8.3-preview20130601.exe&can=2&q=>
  2. Al instalarlo, cuando pregunta para usar Git Bash o Command Prompt, elegir Command Prompt.
  3. Cuando pregunta acerca de cómo realizar el checkout de Windows y Linux, elegir la opción por defecto (la de más arriba).
2. Ejecutar Git Bash. Para ello ir a Start y buscar "Git Bash" (si se trabaja en Windows 7).
3. Crear claves de SSH. Dentro de Git Bash, ejecutar el comando:

**ssh-keygen -t rsa.**

Elegir todas las opciones por defecto cuando se lo pida (apretar Enter cada vez). Esta acción crea un par de claves ssh en la carpeta \$USER/.ssh (siendo \$USER la carpeta del usuario, "C:\Users\usuario" en Windows 7 o "C:\Documents and Settings\usuario" en Windows XP). Estas claves tienen el nombre "id\_rsa" y "id\_rsa.pub" (para la clave privada y pública respectivamente). Si se pregunta para sobrescribir tales claves, entonces ya se cuenta con una clave ssh, por lo tanto se puede seguir al siguiente paso.

4. Copiar la clave pública generada. Para ello ir a la carpeta \$USER/.ssh, abrir "id\_rsa.pub" con notepad, y copiar todo el texto de dentro del archivo.
5. Setear claves en la cuenta de CloudForge. Para ello ir a la cuenta CloudForge, e ir a My Settings->Keys->Add SSH Key. Ingresar la clave que se tiene copiada en el portapapeles, e ingresar un nombre descriptivo para ella (se sugiere identificar tales claves si se va a acceder a los repositorios en distintas máquinas).
6. Setear nombre y email en git. Para ello abrir Git Bash y ejecutar los siguientes comandos:

**git config --global user.name [nombre]**

**git config --global user.email [email]**

Donde **[nombre]** es el nombre del integrante, y **[email]** su email.

Por más información ver: <http://guides.beanstalkapp.com/version-control/git-on-windows.html>.

Si se quiere usar Putty y claves .ppk, se deberán seguir los pasos de arriba. Sin embargo se sugiere utilizar OpenSSH, y en este documento se asumirá que se está utilizando OpenSSH.

Si no funciona clonar el repositorio con las herramientas de Windows descritas abajo, probar copiar el archivo "~/.ssh/id\_rsa" a "C:\Program

Files\Git\.ssh\ir\_rsa" (sustituir "C:\Program Files\Git" por el directorio donde se instaló git).

## 2.5. Herramientas de Git

A continuación se describen las posibles herramientas para poder utilizar git, acceder a repositorios, etc.

Para Linux, se recomienda utilizar la línea de comando o Git Cola si no se necesitan realizar demasiadas operaciones sobre el repositorio (por ejemplo sólo para hacer pull). Si se necesitan hacer acciones más complejas, se recomienda utilizar GitEye, ya que ofrece más funcionalidades que Git Cola y ofrece una mejor interfaz gráfica. Sin embargo GitEye requiere de más configuraciones, por lo tanto si es posible se debería utilizar Git Cola, al menos que el integrante quisiera experimentar con GitEye.

Para Windows se recomienda utilizar TortoiseGit para realizar acciones livianas y rápidas sobre el repositorio sin la necesidad de ejecutar otra aplicación (por ejemplo si solamente se quiere actualizar el repositorio local). Para operaciones más pesadas y que requieren un mayor control se recomienda utilizar SourceTree o GitEye, por ejemplo para commitear sobre el repositorio de elementos del sistema, o para visualizar el histórico del repositorio, etc. Para Windows es preferible utilizar SourceTree a GitEye, ya que GitEye necesita más configuraciones que SourceTree.

En ambos casos, se recomienda utilizar la integración con git de Aptana Studio 3 para llevar a cabo la implementación del producto.

### 2.5.1. Git CLI (Command Line Interface):

Sistema Operativo: Linux & Windows.

Descripción: Esta herramienta permite trabajar con git mediante la línea de comandos.

Instalación: Si se siguieron los pasos descritos anteriormente para instalar git en su máquina (Linux o Windows), entonces git ya puede ser utilizado mediante CLI. Para utilizarlo en Linux basta con abrir una terminal y posicionarse sobre la carpeta que contiene el repositorio donde se quiera trabajar. Para utilizarlo en Windows se deberá abrir Git Bash y seguir el mismo procedimiento.

Guías:

Los siguientes links muestran los posibles comandos a utilizar en git mediante la línea de comandos y lo que realizan. Es recomendable leer las guías de git mostradas previamente, ya que también contienen ejemplos del uso de git en línea de comandos.

[http://ndpsoftware.com/git-cheatsheet.html#loc=local\\_repo](http://ndpsoftware.com/git-cheatsheet.html#loc=local_repo);

[https://na1.salesforce.com/help/doc/en/salesforce\\_git\\_developer\\_cheatsheet.pdf](https://na1.salesforce.com/help/doc/en/salesforce_git_developer_cheatsheet.pdf)

### 2.5.2. Git Cola:

Sistema Operativo: Linux.

Descripción: Git Cola es una GUI que permite trabajar con git. Ofrece una buena interfaz para realizar operaciones comunes (como Commit, Stage, etc), y es una de las mejores herramientas para utilizar en Linux.

Instalación: Abrir una terminal y ejecutar el comando **apt-get install git-cola**

Guías:

<http://git-cola.github.io/share/doc/git-cola/html/index.html>



### 2.5.3. Source Tree:

Sistema Operativo: Windows.

Descripción: Source Tree es una aplicación que permite trabajar con git en Windows. Tiene muchas funcionalidades, es muy vistosa, y es fácil aprender a usarla. Si se necesita tener un control sobre el repositorio donde se trabaja, o si se necesitan realizar varias acciones sobre él, o si se quiere tener un acceso fácil y rápido al histórico, a los branches, tags, etc del repositorio, es la herramienta recomendada para Windows.

Instalación: Ir al siguiente sitio y descargar SourceTree: <https://www.atlassian.com/software/sourcetree/overview> .

Instalarlo, y cuando dé la opción, elegir como nombre de usuario y email los mismos utilizados cuando se configuró git. Cuando de a elegir entre Putty y OpenSSH, elegir OpenSSH.

Una vez instalado, abrir la aplicación, e ir a Tools->Options, y en SSH-Client elegir OpenSSH. En la sección "SSH Key", seleccionar la clave privada creada previamente (tal clave se encuentra en "\$USER/.ssh/id\_rsa").

Después de ser configurado, es necesario registrar la copia de Sourcetree. Para ello se deberá ir a "File"->"Registration...", y seleccionar "Register Now!". Luego se deberá ingresar el email, y luego se deberán ingresar los datos que indiquen en pantalla. Luego se podrá registrar. Otra opción es registrarse de forma online (en <https://id.atlassian.com/profile/signUp.action?application=mac&continue=https://my.atlassian.com/license/sourcetree>), y en Sourcetree, cuando se ingrese el email, se podrá acceder a la cuenta con el password creado.

Guías:

No tiene. Sin embargo es relativamente fácil aprender como utilizarlo. Si realmente es necesario se puede consultar el siguiente video: <http://www.youtube.com/watch?v=QnkiKrruJiE>

### 2.5.4. Tortoise Git:

Sistema Operativo: Windows.

Descripción: TortoiseGit es una herramienta para utilizar git. Esta herramienta es muy similar a TortoiseSVN. Esta herramienta permite realizar operaciones usando el menú de contexto del explorador de Windows. Es decir, se puede estar posicionado sobre una carpeta en el explorador de Windows, y al hacer click derecho sobre un archivo o sobre la pantalla, se habilita un sub-menú de TortoiseGit para realizar operaciones. Esta herramienta es recomendada si no se quiere tener mucho control sobre el repositorio en sí, y si solamente se quiere realizar operaciones de git rápidamente sin tener que ejecutar una aplicación para hacerlo. Por ejemplo, si solamente se quiere hacer "pull" de datos nuevos, se recomienda utilizar esta herramienta. Por este motivo es recomendable que se utilice esta herramienta para que los integrantes del equipo puedan acceder al repositorio de elementos del sistema.

Instalación: Ir al siguiente sitio y descargarlo <https://code.google.com/p/tortoisegit/> . Una vez descargado, instalarlo. Cuando dé la opción, elegir OpenSSH.

Guías:

<https://code.google.com/p/tortoisegit/wiki/Screenshots>

<https://code.google.com/p/tortoisegit/wiki/FAQ>

### 2.5.5. GitEye:

Sistema Operativo: Linux & Windows.

Descripción: Esta es una herramienta muy completa que puede ser utilizada en Linux y en Windows. Pertenecce a Collab.NET, por lo que ofrece integración con CloudForge y si es necesario TeamForge. Su implementación es basada en Eclipse, y cuenta con soporte de EGit y JGit, por lo tanto implementada utilizando Egit y JGit

Instalación: Ir al sitio <http://www.collab.net/downloads/giteye> y descargar el archivo de instalación correspondiente.

Una vez descargado, se deberá descomprimir la carpeta descargada en un lugar a elección. Una vez hecho se podrá ejecutar GitEye.

Luego de ejecutarlo, se deberán seguir estos pasos:

- Apretar el botón "Learn More" a la derecha de "Registration Required".
- En la nueva pantalla seleccionar "Connect to Your Account" y presionar "Add Connection". Debajo de "Add New Site" elegir CloudForge.
- En la siguiente pantalla, seleccionar una descripción de su elección, y seleccionar Next.
- Escribir su nombre de usuario de su cuenta de CloudForge, su contraseña, y el domain pis2013gr11. Al finalizar apretar "Finish".
- Luego, ir a Window->Preferences->General->Network Connections->SSH2. Seleccionar "Add Private Key"; luego ir al directorio ~/.ssh/ y seleccionar la clave "id\_rsa".

Guías:

Guía de GitEye: <https://ctf.open.collab.net/sf/projects/giteye/>

Guía de EGit: [http://wiki.eclipse.org/EGit/User\\_Guide](http://wiki.eclipse.org/EGit/User_Guide)

### **2.5.6. Integración git de Aptana Studio 3**

Sistema Operativo: Linux & Windows.

Descripción: Aptana Studio 3 incluye integración con git, para de esta forma poder trabajar con el repositorio y realizar operaciones sobre él desde el mismo ambiente donde se desarrolla. Es la opción recomendada para utilizar el repositorio de desarrollo y prototipo.

Instalación: Una vez instalado el Aptana Studio 3, no es necesario instalar nada más. Lo único necesario es que al instalarlo, se setee el workspace en la carpeta padre donde se quiere tener el o los repositorios para trabajar.

Guías:

<https://wiki.appcelerator.org/display/tis/Git>

### **2.6. Clonar un repositorio git**

Una vez instalado git, y una vez instalada la herramienta correspondiente, es posible clonar un repositorio remoto para poder accederlo de forma local, o para trabajar sobre él.

Antes que nada, se deberá obtener la url del proyecto a clonar. Para ello, se deberá ir a la cuenta de CloudForge; después se deberá ir al proyecto que contiene el repositorio que se quiere clonar, y en la página principal de tal proyecto se deberá copiar la url de SSH que aparece en pantalla.

Si se va a clonar un repositorio utilizando la herramienta GitEye, no es necesario obtener la URL del repositorio previamente.

#### **2.6.1. Git CLI**

1. Abrir una terminal/GitBash (si se trabaja en un ambiente Linux o Windows respectivamente).
2. Posicionarse sobre la carpeta de trabajo del proyecto utilizando el comando **cd** (e.g "cd ~/pis").
3. Ejecutar el siguiente comando:  
git clone [url]  
Donde **[url]** es la url obtenida previamente. Esto creará una carpeta que contiene el repositorio clonado.
4. Si se quiere crear una carpeta con un nombre en particular, ejecutar el siguiente comando:  
git clone [url] [carpeta]  
Donde **[carpeta]** es el nombre de la carpeta que se quiere.

#### **4.1.1. Git Cola**

1. Ejecutar git-cola
2. Cuando aparezca la ventana inicial, seleccionar "Clone ...".
3. Pegar la url obtenida previamente y seleccionar "Ok".
4. Elegir la carpeta donde se quiere clonar tal repositorio.

#### **4.1.1. SourceTree**

1. Ejecutar SourceTree.
2. Apretar el botón "Clone/New".
3. En la nueva pantalla, pegar la url obtenida previamente, y seleccionar la carpeta local donde se quiere clonar el repositorio.

#### **3.1.1. TortoiseGit**

1. Ir a la carpeta donde se quiere clonar el repositorio.
2. Hacer click derecho sobre el explorador de Windows, y seleccionar "Git. Clone...".
3. Pegar la url obtenida previamente y seleccionar "Ok".

#### **3.1.2. GitEye**

1. Ejecutar GitEye
2. Ir a "Git Repositories" en el tab de la izquierda.
3. Seleccionar "Clone a Git repository".
4. En la nueva pantalla seleccionar "CloudForge".
5. En la siguiente pantalla se listarán todos los repositorios de su cuenta de CloudForge. Seleccionar el que se quiere clonar, y seleccionar "SSH" en el campo "Clone via...".
6. Apretar "Next". Si aparece el cartel "The authenticity of host..." seleccionar Yes.
7. En la nueva pantalla seleccionar todos los branches/etc, y presionar "Next".
8. En la nueva pantalla elegir la carpeta donde se va a clonar el repositorio, y seleccionar Finish.

9. Una vez clonado el repositorio, se va a mostrar una ventana que indica que se tiene que agregar un archivo "proyect". Dejar la opción "Agregar a .gitignore" desmarcada, y apretar "Ok"
10. Luego, ir a la carpeta del repositorio (se puede usar el File Explorer de GitEye, que viene en el tab de "Git Repositories"), y abrir el archivo ".git/info/exclude".
11. En tal archivo agregar "./proyect" en una nueva línea. Guardar
12. Una vez guardado, se deberá reiniciar GitEye.  
Se deberá realizar esto para que el archivo ".proyecto" no sea commitado ni tenido en cuenta por git.

### **3.1.3. Aptana Studio 3**

1. Ejecutar Aptana Studio 3.
2. Ir a File->Import...
3. En la nueva ventana seleccionar la carpeta Git, y seleccionar la opción que aparece debajo de ella.
4. En la siguiente ventana, pegar la url obtenida previamente, y luego elegir la carpeta donde se quiere clonar el repositorio.

## **4. Descripción del uso del Ambiente de Trabajo**

Para poder utilizar el ambiente de trabajo, es necesario tener una cuenta de Gmail. Una vez que se tiene tal cuenta, el SCMR se encargará de compartir la carpeta "PIS" (que corresponde a tal ambiente) con él.

### **4.1. Herramientas**

A continuación se describen las herramientas a utilizar en el ambiente de trabajo, y se describe cómo utilizarlas:

#### **4.1.1. Cacao**

Sitio: <https://cacao.com/lang/es/>

Cacao es una herramienta para diseñar diagramas, en especial diagramas UML.

Para poder utilizarla, es necesario acceder al sitio y registrar una cuenta.

Para que todos los integrantes del equipo puedan acceder a tales diagramas, y colaborar en ellos, es necesario de que tales diagramas estén guardados en Google Drive y no en Cacao en sí. Esta guía indica como asociar diagramas de Cacao a Google Drive: <http://blog.cacao.com/google-drive-manual/> .

Para más información acerca de cómo utilizar Cacao ir a <https://cacao.com/lang/en/tour> .

#### **4.1.2. Ganttter**

Sitio: <http://gantter.com/>

Ganttter es una herramienta que permite crear, modificar, y colaborar con diagramas Gantt de forma online.

Esta herramienta será utilizada para trabajar sobre el Plan de Iteración, y para poder accederlo de forma online. Para ello es necesario que su cuenta de Google esté configurada para que pueda utilizar tal herramienta.

Para ello deberá seguir los siguientes pasos:

1. Loguearse a su cuenta de Google, e ir a Google Drive

2. Ir a "Configuración" (hacer click sobre el engranaje arriba a la derecha), y luego a "Administrar Aplicaciones"
3. En la nueva ventana, seleccionar "Conectar más aplicaciones"
4. En la nueva ventana, posicionarse sobre el campo de búsqueda y escribir "Ganttter"
5. Al terminar la búsqueda deberá aparecer la aplicación "Ganttter for Google Drive". Apretar el botón de conectar para poder conectar tal aplicación a su cuenta

Una vez que haya conectado Ganttter a su cuenta de Google Drive, podrá acceder a los diagramas de Gantt del proyecto en Google Drive solamente haciendo click sobre ellos.

## 4.2. Sugerencias

A continuación se detallan algunas sugerencias para el uso del ambiente de trabajo:

- Para la elaboración de documentos, si éste necesita de la colaboración de muchos, o si es necesario que el integrante responsable de él necesite accederlo de varias máquinas, se sugiere que trabaje con tal documento en Google Docs (ver sección "Definición del Ambiente Controlado" para más información).

Google Docs tiene problemas de compatibilidad con ODF, por lo tanto se sugiere el siguiente procedimiento para poder trabajar con tal herramienta:

- Primero, empezar con un documento de extensión .odt base. Éste puede ser una versión anterior del documento, o si el documento no ha sido creado todavía puede ser la plantilla del mismo.
- Ir a Google Drive y crear un documento de Google Doc nuevo. En el documento, ir a Archivo->Abrir.
- En la nueva ventana seleccionar "Cargar", y seleccionar el archivo de extensión .odt descrito previamente
- Trabajar sobre el documento.
- Una vez terminado, crear una copia del archivo .odt original, y copiar las secciones modificadas/creadas del archivo en Google Docs a este archivo .odt nuevo.
- Una vez copiadas, ajustar el formato del archivo .odt adecuadamente.

De esta forma se minimiza el impacto que tal falta de incompatibilidad pueda tener sobre el trabajo de los integrantes.

- Para la elaboración de diagramas UML, se recomienda crear el diagrama de la forma descrita en la sección "Cacao" de este documento. Una vez terminado el diagrama, se deberá exportar con formato PNG. Cuando se necesita crear una versión nueva del diagrama, se recomienda o guardar el histórico de este diagrama en Cacao (si es posible), o guardar este diagrama en una nueva carpeta llamada "Otros Diagramas" en la carpeta de Google Drive correspondiente.

## 5. Responsabilidades

El responsable de la integridad de los repositorios es el SCMR.

Para el caso del código fuente el Responsable de Integración es responsable en buena medida del manejo de las ramas de desarrollo, de integrar nuevas funcionalidades a la rama principal de desarrollo y demás. A su vez el SCMR es responsable de definir procedimientos para el manejo de tal repositorio cuando lo crea necesario, y es responsable de asegurar que tales procedimientos se sigan.

Para el caso del repositorio del prototipo, los Especialistas Técnicos son responsables del uso del repositorio y su integridad, aunque últimamente ante cualquier inconsistencia el SCMR se hará cargo de asegurar su integridad.

Cada integrante del equipo es responsable del correcto manejo del ambiente de control y del uso de las herramientas proporcionadas previamente. Ante cualquier duda, el integrante podrá comunicarse con el SCMR.

Ningún integrante del equipo, que no haya sido autorizado por el SCMR, deberá, en ninguna forma, modificar el repositorio de elementos del sistema ni el repositorio de línea base. El SCMR puede autorizar a algún integrante del equipo (como por ejemplo el SQAR) para que pueda modificar el repositorio de elementos, pero el SCMR es el único responsable de modificar el repositorio de línea base. Para los integrantes del equipo que tienen acceso a estos dos repositorios, pero no tienen autorización para modificarlos, deberán ser cuidadosos al momento de utilizar las herramientas descritas anteriormente para trabajar sobre tales repositorios. Ante cualquier duda o inconsistencia el SCMR deberá darles soporte para que puedan seguir trabajando. Se recomienda solamente hacer "pull" de estos repositorios, y no realizar ninguna operación más.

Por más información, ver el Plan de Configuración.

No es obligatorio que los integrantes del equipo deban utilizar las herramientas descritas anteriormente. Sin embargo, si utilizan otras herramientas, es responsabilidad de ellos de asegurarse que las están utilizando de forma correcta, y si encuentran problemas, de resolverlos. El SCMR es responsable de dar soporte únicamente para el uso de las herramientas proporcionadas previamente en este documento.

Todo integrante del equipo es responsable de no modificar la estructura de directorios general de la carpeta compartida de Google Drive (como ha sido descrita en la sección "Definición del Ambiente Controlado"). A su vez solamente el SCMR y la Administradora son responsables de manipular los archivos y carpetas del Plan de Iteración y del Registro de Horas en Google Drive, a menos que se tenga autorización de uno de ellos para hacerlo (en cual caso es responsabilidad de ellos asegurar de que tal manipulación se esté realizando correctamente).

No es obligatorio que los integrantes del equipo deban seguir los procedimientos sugeridos en la sección "Sugerencias", pero es muy recomendable que lo hagan.

Acerca de los diagramas UML, no hay responsabilidades estrictas sobre compartir los diagramas en Google Drive, pero es una recomendación que ayuda a que cada integrante tenga acceso a ellos y permite tener un ambiente único de trabajo, por lo tanto se sugiere hacerlo, y el SCMR se encargará de sugerirlo y asegurarse de que sea hecho cada vez que lo crea necesario.