

Tecnologías Web: Cliente Herramientas.

Herramientas para las tareas de desarrollo

- Editor de texto
- Navegador
- Gestión de paquetes: npm, brower
 - **Gestión de los paquetes en el lado del cliente**
- Control de versiones: Git
- Generación del proyecto: Yeoman
 - Crear la estructura del proyecto
- Automatización de tareas Gulp, Grunt



Yeoman

- Generación del proyecto rápida del proyecto:
 - Crear la estructura del proyecto
- Instalación:
npm install -g yo
- La instalación admite paquetes que agregan otras herramientas necesarias para el proyecto
- Agregar generadores disponibles:

```
npm install -g yo //instalar yeoman
npm install -g generator-webapp // Instalar el generador
mkdir /home/tuUsuario/nuevoProyecto // Crear la carpeta del proyecto
cd /home/tuUsuario/nuevoProyecto // Entrar en la carpeta proyecto
yo webapp // Ejecutar el generador webapp
```



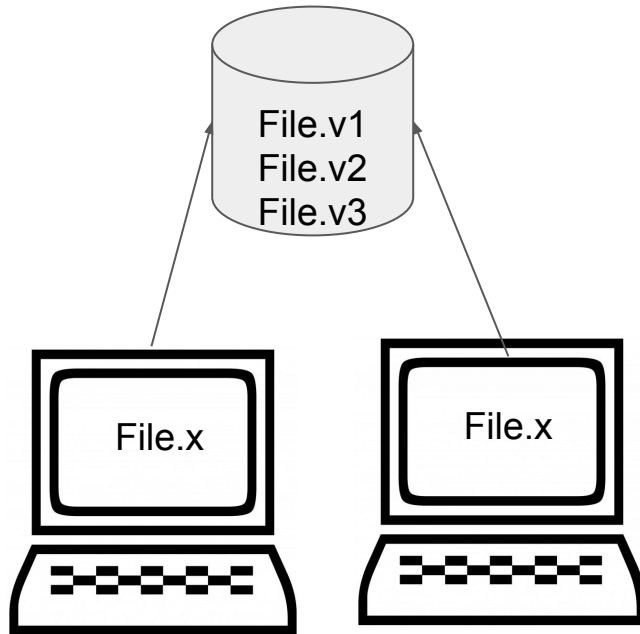
Herramientas para el desarrollo de videojuegos

- Control de versiones:
 - Gestión de los cambios en un conjunto de archivos, quién los ha realizado, estado actual del proyecto.
- Tipos de sistemas de control de versiones:
 - Sistemas centralizados: el proyecto está en un único servidor, el usuario dispone de los archivos con los que está trabajando
 - Subversion, Perforce
 - Sistemas distribuidos: Cada miembro del equipo mantiene una versión del proyecto que también puede estar en un servidor.
 - Git



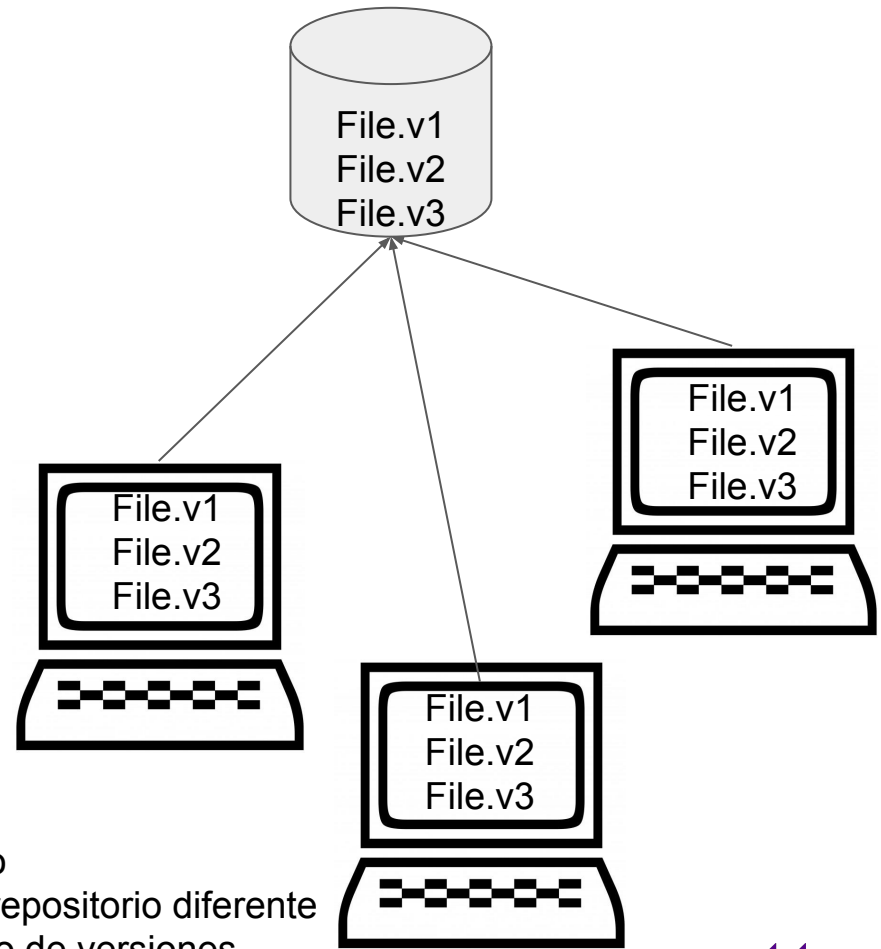
Herramientas para el desarrollo de videojuegos

- Sistema Centralizado vs Distribuido:



Sencillo

Nº de versión para cada etapa
del proyecto



Rápido

Cada repositorio diferente
número de versiones



Herramientas para el desarrollo de videojuegos

- Configuración **git config**
 - Se edita el fichero de parámetros de configuración de Git del sistema, del usuario o
 - **git config --list** Muestra la configuración

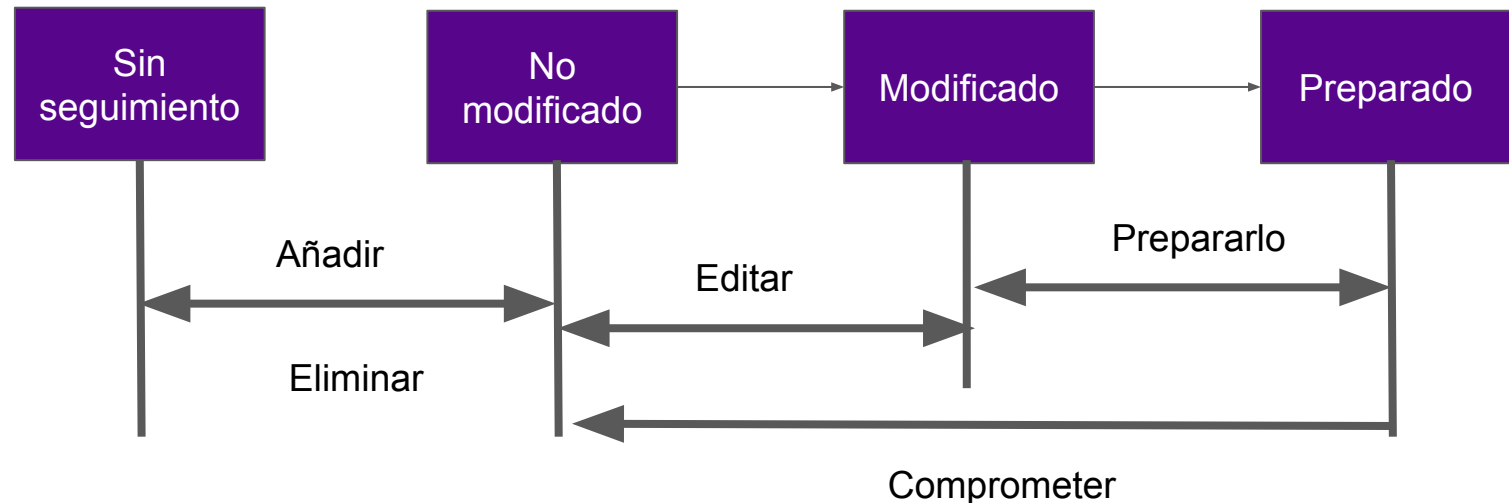
Sistema	<code>git config --system</code>	<code>\$HOME /etc/gitconfig</code>
Usuario	<code>git config --global</code>	
Repositorio	<code>git config --local</code>	

```
git config --list
user.name=isanchezberriel
user.email=isanchez@ull.es
merge.tool=vimdiff
```



Herramientas para el desarrollo de videojuegos

- Estados de los archivos:
 - El repositorio se considera una instantánea del conjunto de archivos.
 - Archivos en seguimiento: están en la instantánea
 - modificado, preparado, no modificado
 - Archivos sin seguimiento: están en el repositorio no está preparado ni estaba en la instantánea



Herramientas para el desarrollo de videojuegos

- Crear repositorio, en el directorio del proyecto:
 - `git init`

```
$ git init
Initialized empty Git repository in
C:/Users/Usuario/Documents/FDV/ejemplosgit/.git/

Usuario@DESKTOP-PICFGN4 MINGW64 ~/Documents/FDV/ejemplosgit
(master)
```

- Averiguar el estado de los archivos:
 - `git status`

```
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to
track)
```



Herramientas para el desarrollo de videojuegos

- Añadir archivos al área de preparación
 - `git add`

```
$ g$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ejemplo.txt
nothing added to commit but untracked files present (use "git add" to
track)

$ git add ejemplo.txt

$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   ejemplo.txt
```



Herramientas para el desarrollo de videojuegos

- Confirmar cambios: registra el área de preparación
 - `git commit`

```
$ git commit
[master (root-commit) 49b2753] Primer fichero
1 file changed, 8 insertions(+)
create mode 100644 ejemplo.txt
```

```
$ git status
On branch master
nothing to commit, working tree clean
```

```
$ git commit
On branch master
Untracked files:
  ejemplo2.txt

nothing added to commit but untracked files present
```



Herramientas para el desarrollo de videojuegos

- Descartar cambios en el área de archivos preparados
 - `git restore --staged nombre_archivo`

```
$ git restore --staged ejemplo2.txt
```

```
$ git status
```

```
On branch master
```

```
Untracked files:
```

```
  (use "git add <file>..." to include in what will be  
  committed)
```

```
    ejemplo2.txt
```

```
nothing added to commit but untracked files present (use "git  
add" to track)
```



Herramientas para el desarrollo de videojuegos

- Descartar cambios de un fichero
 - `git reset --nombre_archivo`

```
$ git restore --staged ejemplo2.txt
```

```
$ git status
```

```
On branch master
```

```
Untracked files:
```

```
  (use "git add <file>..." to include in what will be  
  committed)
```

```
    ejemplo2.txt
```

```
nothing added to commit but untracked files present (use "git  
add" to track)
```



Herramientas para el desarrollo de videojuegos

- Mostrar el historial de cambios
 - `git log`

```
$ git log
commit 6a00bfe9985abdf9b579e07e1c35efb0a80c76a3 (HEAD ->
master)
Author: isanchezberriel <isanchez@ull.es>
Date:   Tue Sep 24 22:31:13 2019 +0100
    Modifico

commit d7974ac2e3f2685e3d655a1159d10f54872ecd18
Author: isanchezberriel <isanchez@ull.es>
Date:   Tue Sep 24 22:28:40 2019 +0100
    Fichero 2

commit 49b2753028a3d17c01c200df6e6a308b26a8e93e
Author: isanchezberriel <isanchez@ull.es>
Date:   Tue Sep 24 21:38:57 2019 +0100
    Primer fichero
```



Herramientas para el desarrollo de videojuegos

- Eliminar archivos

- `git rm`

```
$ git rm ejemplo2.txt
rm 'ejemplo2.txt'
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted: ejemplo2.txt
```

- Eliminar archivos

- `git mv`

```
$ git mv ejemplo.txt ejemplo3.txt
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed: ejemplo.txt -> ejemplo3.txt
```



Herramientas para el desarrollo de videojuegos

- Mostrar los repositorios remotos
 - `git remote [-v]`
 - `git remote show [nombre]`
- Añadir repositorios remotos
 - `git remote add [nombre] [url]`
- Recuperar repositorios remotos
 - `git clone [repositorio remoto]`
- Recuperar archivos añadidos a repositorios remotos
 - `git fetch [nombre_repositorio_remoto]`
 - `git fetch [origin]`
- Recuperar y fusionar una rama remota con la rama actual
 - `git pull`
- Enviar repositorio local actual a repositorio remoto
 - `git push`



Herramientas para el desarrollo de videojuegos

- Generar clave para conexión ssh

```
~/.ssh$ ssh-keygen -t rsa -C "isanchez@ull.es"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/isabel/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/isabel/.ssh/id_rsa.
Your public key has been saved in /home/isabel/.ssh/id_rsa.pub.
The key fingerprint is:
...
~/.ssh$ ls -l
total 12
-rw----- 1 isabel isabel 1675 sep 26 09:49 id_rsa
-rw-r--r-- 1 isabel isabel  397 sep 26 09:49 id_rsa.pub
-rw-r--r-- 1 isabel isabel 1110 jun  5 18:29 known_hosts
~/.ssh$ gedit id_rsa.pub

$ ssh -T git@github.com
...
Hi isanchezberriel! You've successfully authenticated, but
GitHub does not provide shell access.
$ git remote add origin
```



Herramientas para el desarrollo de videojuegos

- Ejemplo de conexión a GitHub

```
$ git remote add git@github.com:isanchezberriel/FDVssh.git
$ git remote -v
origin      git@github.com:isanchezberriel/FDVssh.git (fetch)
origin      git@github.com:isanchezberriel/FDVssh.git (push)
$ git push -u origin master
```

```
$ git remote -v
origin      git@github.com:isanchezberriel/FDVssh.git (fetch)
origin      git@github.com:isanchezberriel/FDVssh.git (push)

$ git pull origin master
$ git push -u origin master
```

Valorar si es necesario actualizar nuestro repositorio con cambios previos de otros desarrolladores



Herramientas para el desarrollo de videojuegos

- Etiquetas: se utilizan para marcar versiones
 - `git tag`

```
$ git tag v.1.0.0

$ git tag
v.1.0.0
$ git push origin master --tags
Total 0 (delta 0), reused 0 (delta 0)
To git@github.com:isanchezberriel/FDVssh.git
* [new tag]          v.1.0.0 -> v.1.0.0

$git push origin master --tags
```

- Convenio etiquetas.
 - Pequeño ajuste **v.1.0.1**
 - Cambio menor **v.1.0.1**
 - Cambio Mayor **v.2.0.1**



Herramientas para el desarrollo de videojuegos

- Ramas
 - `git branch [nombre de rama]`
 - `git branch-show`
 - `git push -u [repositorio] [nombre de rama]`
 - `git merge [nombre de rama] -m 'Mensaje'`
 - Fusiona la rama actual con la rama referenciada
 - `git checkout [nombre de rama]`
 - Cambio de rama
 - `git push -u origin [nombre de rama]`
- Fork



Herramientas para el desarrollo de videojuegos

```
$ git branch mirama
$ git show-branch
* [master] Update README.md
! [mirama] Update README.md
--
*+ [master] Update README.md
$ git status
En la rama master
Su rama está actualizada con «origin/master».
$ git checkout mirama
Switched to branch 'mirama'
$ git status
En la rama mirama
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...» para sacar del stage)
    nuevo archivo: archivos/ejemplo3.txt
$git show-branch
! [master] Update README.md
* [mirama] Añadido un nuevo archivo en esta rama. Se ha creado para
mostrar los comandos relacionados con las ramas. Este archivo no está
en la rama master
--
* [mirama] Añadido un nuevo archivo en esta rama. Se ha creado para
mostrar los comandos relacionados con las ramas. Este archivo no está
en la rama master
+* [master] Update README.md
```



Herramientas para el desarrollo de videojuegos

```
$ git branch
  master
* mirama
$ git push -u origin mirama
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 476 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'mirama' on GitHub by
visiting:
remote:
https://github.com/isanchezberriel/FDVssh/pull/new/mirama
remote:
To git@github.com:isanchezberriel/FDVssh.git
 * [new branch]      mirama -> mirama
Branch mirama set up to track remote branch mirama from
origin.
```



Herramientas para el desarrollo de videojuegos

- Descartar archivos para el seguimiento: Los archivos que se listen en el fichero **.gitignore** no se consideran en el seguimiento.
 - Editar el fichero
 - Patrones para **.gitignore**:
 - *
 - ** Múltiples niveles del directorio
 - ! Excepción
- Activar el seguimiento de .gitignore para que todo el equipo descarte los mismos ficheros.



Herramientas para el desarrollo de videojuegos

- **Atributos** en Git para especificar estrategias diferenciadas para algunos ficheros o directorios.
- Permiten especificar cómo comparar ficheros binarios de forma efectiva.
 - **.gitattributes**
- En los casos en que Git no podría llegar a determinarlo por sí mismo, se pueden especificar conversiones a texto y que luego se aplique **diff** para verificar si hay diferencias.
- Activar el seguimiento de **.gitattributes** para que todo el equipo descarte los mismos ficheros.



Gestionar dependencias

- npm: gestión de dependencias en node, se ha extendido su uso en el frontend

`npm init` //en el directorio del proyecto

Crea el archivo `package.json`

- bower: gestión de dependencias en el lado del cliente.
 - Creado por twitter
 - Buscar paquetes en: <https://bower.io/search/>
 - Crea el archivo `bower.json`

```
npm install -g bower
cd mi_proyecto
bower init
bower install <package> --save
```



Tareas repetitivas

- Pre-procesar y minificar CSS
- Unificar y pre-procesar javascript
- Mover ficheros minificados a un directorio
- Preparar pruebas de ejecución y errores
- Recarga automática del navegador
- Optimizar imágenes para la web

Recursos a minificar

- CSS
- JavaScript
- Optimizar imágenes



Gulp

- Herramienta de automatización de tareas repetitivas y/o comunes en el desarrollo del proyecto.
- Gulp convierte los archivos de entrada en flujos de datos en memoria, se minimizan las operaciones de escritura en disco.
- Automatizadores están disponibles para Gulp en forma de plugins
- Se pueden vigilar en tiempo real los archivos que queramos de nuestro proyecto, y ejecutar ciertas tareas de Gulp cada vez que detecte cambios en ellos.
- Todas las tareas que queremos automatizar se programan de forma sencilla en un script que se aloja en la raíz del proyecto:
 - **gulpfile.js**



Gulp

- Instalación: `npm install --global gulp-cli`
- Agregar gulp al proyecto: `npm install gulp --save-dev`
- Fichero gulpfiles.js básico:

```
var gulp = require('gulp');  
// Tareas por defecto  
gulp.task('default', function() {  
    // tareas que queremos que se ejecuten cada vez que hagamos gulp  
});
```

- `gulp.src().pipe(gulp.dest())` Crea un flujo de la fuente al destino.

```
var gulp = require('gulp');  
var concatCss = require('gulp-concat-css'); //paquete a utilizar  
  
gulp.task("default", function() { //tarea por defecto  
    gulp.src('./src/html/*.html').pipe(gulp.dest('dist'));  
    gulp.src('./src/css/*.css').pipe(concatCss("super.css")).pipe(gulp.dest('dist'));  
    gulp.src("./src/js/*.js").pipe(gulp.dest("dist"));  
});
```



BrowerSyn: browsersync.io

- Permite lanzar diferentes navegadores e ir verificando cambios en proyecto en tiempo real.
- Podemos especificar a qué cambios responderá la recarga
- Podemos automatizar
- Instalación: `npm install -g browser-sync`
- Inicializar: `browser-sync start --server --files "css/*.css"`

```
var gulp = require('gulp');
var browserSync = require('browser-sync').create(); //paquete a utilizar

gulp.task('browser-sync', function() {
  browserSync.init({
    server: {
      baseDir: "./"
    }
  });
});
gulp.watch("*.html").on("change", browserSync.reload);
```



Minificar:

- Permite eliminar caracteres superfluos del archivo
- Paquete para css: **gulp-minify-css**
- Instalación **npm install --save-dev gulp-minify-css**

```
var gulp = require('gulp');
var browserSync = require('browser-sync').create();
var minifyCss = require('gulp-minify-css');

gulp.task('styles', function(){
  gulp.src(['src/styles/**/*.css'])
    .pipe(minifyCss())
    .pipe(gulp.dest('dist/styles'))
    .pipe(browserSync.stream());
});

gulp.task('default', function(){
  browserSync.init({
    server: './'
  });
  gulp.watch('*.html', browserSync.reload);
});
```



Sourcesmaps

- Sourcesmaps nos permiten ver ficheros minificados en el navegador como fueron creados originalmente
- Paquete: **gulp-sourcesmaps**
- Instalar: **npm install --save-dev gulp-sourcesmaps**

```
var gulp = require('gulp');
var browserSync = require('browser-sync').create();
var minifyCss = require('gulp-minify-css');
var sourcesmaps = require('gulp-sourcesmaps')

gulp.task('styles', function(){
  gulp.src(['src/styles/**/*.css']).
    .pipe(sourcesmaps.init())
    .pipe(minifyCss())
    .pipe(sourcesmaps.write())
    .pipe(gulp.dest('dist/styles'))
    .pipe(browserSync.stream());
});

gulp.task('default', function(){
  browserSync.init({
    server: './'});
  gulp.task('styles', function(){
    gulp.src(['src/styles/**/*.css']).
      .pipe(sourcesmaps.init())
      .pipe(minifyCss())
      .pipe(sourcesmaps.write())
      .pipe(gulp.dest('dist/styles'))
      .pipe(browserSync.stream());
  });
  browserSync.serve();
});
```



