

# Gestión de la construcción

---

Para la gestión de la construcción se usan combinadamente la herramienta Bower como gestor de dependencias y Grunt para automatizar la generación y despliegue de la aplicación.

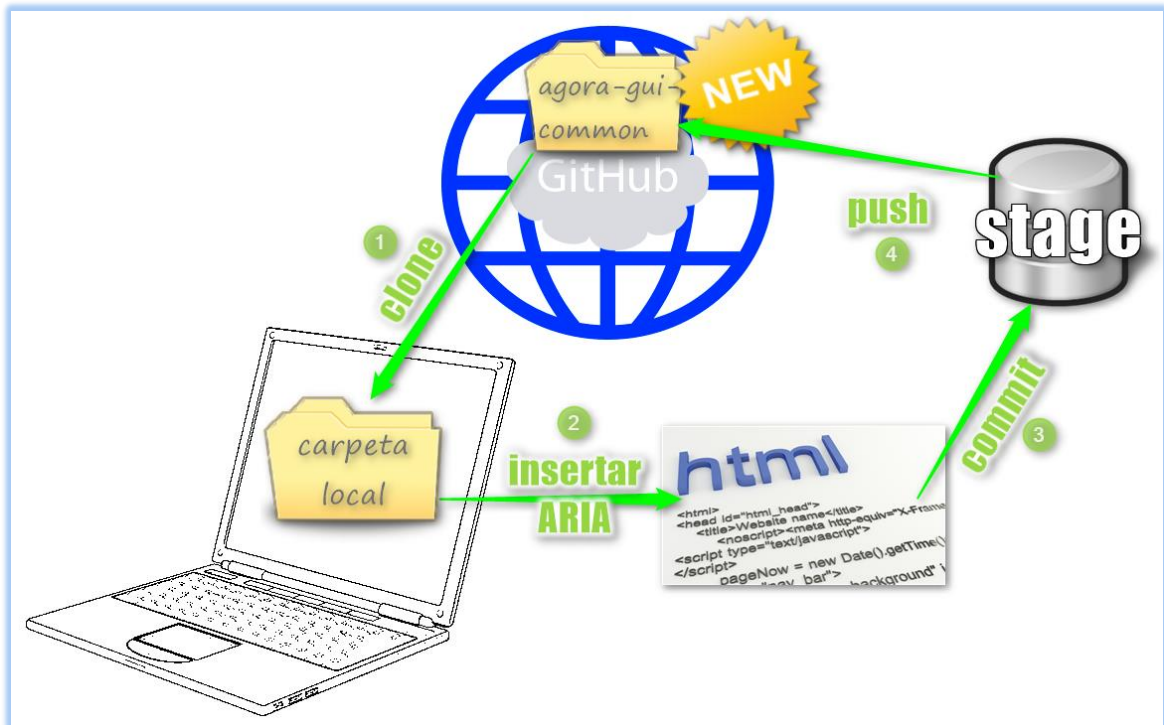
Dado que Bower obtiene los archivos de código desde GitHub, una de las problemáticas encontradas es que el flujo de trabajo se complica ligeramente, ya que para poder desplegar modificaciones deben haberse subido primero a GitHub. Esto implica que a veces puedes subir código no comprobado a GitHub para posteriormente, una vez desplegado y pudiendo comprobar el comportamiento, encontrar un error y tener que acometer el proceso completo de nuevo.

Básicamente el proceso completo se centra en el script de construcción “build.sh” contenido dentro del sistema de la máquina virtual de Agora. Este script se encarga de gestionar las dependencias usando Bower y efectuar la construcción con Grunt, además coloca todos los archivos finales en la ruta correcta lista para ser consultada externamente desde el navegador.

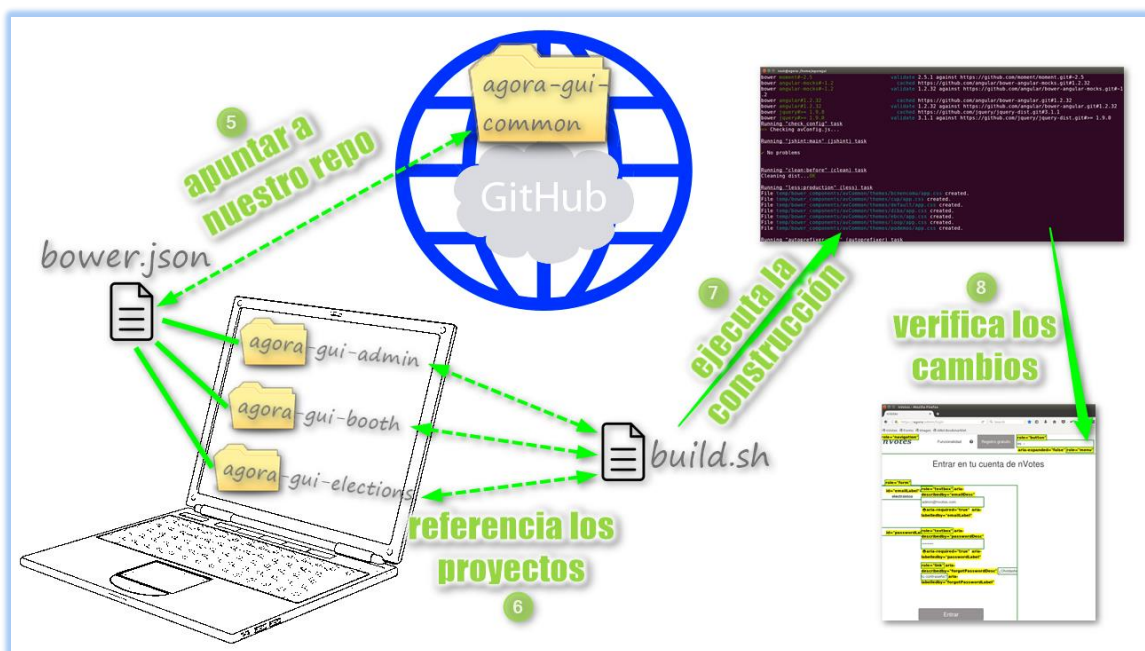
Previo a la ejecución del constructor, hay que configurar Bower en cada uno de los subproyectos de interfaz de usuario (agora-gui-admin, agora-gui-booth y agora-gui-elections) para especificar que obtengan el módulo común (agora-gui-common) desde nuestro repositorio remoto, que es donde están los cambios y modificaciones con respecto al sistema original.

Tras ello, se ejecuta el script de construcción “build.sh” que irá llamando a cada uno de esos subproyectos y ejecutando Bower (que ya apunta a nuestro repositorio de código) para gestionar y descargar las dependencias, incluyendo el contenido de nuestro repositorio remoto, y Grunt para construir la estructura de la web.

A continuación se muestra un esquema del proceso a seguir, esperando conseguir una mayor comprensión del flujo de trabajo en la construcción:



- 1) Se debe tener el repositorio agora-gui-common del repositorio remoto en local, en principio mediante un “git clone”.
- 2) Una vez con el sistema de interfaz común en local, acceder a los archivos de código fuente e insertar los cambios deseados. En nuestro caso inyectar etiquetas ARIA en los distintos elementos web para mejorar la accesibilidad.
- 3) Realizar el commit de los cambios a la zona intermedia.
- 4) Subir los cambios a nuestro GitHub remoto mediante un “git push”.



- 5) Dentro de cada repositorio de interfaz de la máquina virtual de AgoraVoting, editar el archivo “bower.json” y especificar la dependencia de la variable “avCommon” hacia nuestro repositorio.
- 6) Una vez todos los repositorios de interfaz tienen referenciado nuestro repositorio común con los cambios aplicados, referenciar correctamente los subproyectos de interfaz de usuario en el script de construcción “build.sh” y las rutas donde debe acabar el sistema construido.
- 7) Una vez configurados los proyectos y el script constructor, ejecutar este último. Comenzará a gestionar las dependencias y a realizar la construcción del sistema en las rutas especificadas en el paso anterior.
- 8) Si todo ha ido bien, deberías poder acceder al servidor web para comprobar que se ha construido la web con los cambios efectuados.

Si se desea implementar otro cambio, se debe comenzar el proceso desde el paso 2, o sea, omitiendo el paso de volver a clonar el repositorio remoto.

Como ventaja principal, al usar Bower y Grunt, una vez realizada una modificación en el sistema se puede construir y hacer un despliegue con bastante facilidad, tan solo ejecutando el script de construcción.

Entre los problemas encontrados, Bower no se puede ejecutar como superusuario, pero sin embargo a veces requiere la creación de alguna nueva carpeta en el sistema, por lo que tras varios problemas debido a ello, se optó por añadir al sistema una instrucción en la configuración de Bower que permitiera la ejecución como superusuario, y finalmente efectuar la construcción siempre como superusuario.

Otro de los principales inconvenientes fue la localización exacta de los archivos de código fuente, ya que dentro del tremendo enmarañado de la estructura heredada de AgoraVoting, no existe una documentación que concrete la estructura o arquitectura específica, y esto complica en gran medida la implementación de nuevo código o modificaciones ya que se pierde mucho tiempo en localizar el archivo específico que debe ser modificado.

Fuente de imágenes:

-Icono ‘Directorio’: [MiRincon-Misaani.blogspot.com.es](http://MiRincon-Misaani.blogspot.com.es)

-Icono ‘Archivo de texto’: [lconfinder.com](http://lconfinder.com)

-Icono ‘Base de datos’: [lconfinder.com](http://lconfinder.com)

-Icono ‘Portátil’: [Dibujos.net](http://Dibujos.net)

-Imagen ‘Código HTML’: [Ciudadano2cero.com](http://Ciudadano2cero.com)

