



UNIVERSIDAD
NACIONAL
DE LA PLATA

SEDICI

REPOSITORIO INSTITUCIONAL DE LA UNLP



Capacitación IUPA

Clase 2

Pablo de Albuquerque, Santiago Tettamanti, Ariel Lira
{pablo, santit, alira}@sedici.unlp.edu.ar

PREBI
prebi.unlp.edu.ar



Esta obra está bajo una [Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/).

27 de Abril de 2022

Índice

- Repaso de actividades de la clase 1
- Introducción a git y herramientas complementarias
- Buenas prácticas de versionado del código fuente de DSpace y sus configuraciones
- Backend
 - Configuración de dspace
 - Archivos de configuración
 - Parámetros básicos
- Frontend
 - Personalización del frontend - Temas
 - Internacionalización
- Actividades clase 2



Repaso clase anterior

- Uso de docker
 - Instalación local del entorno de DSpace
 - Problemas?



Versionado de DSpace con Git



Versionado del código - Git

Características

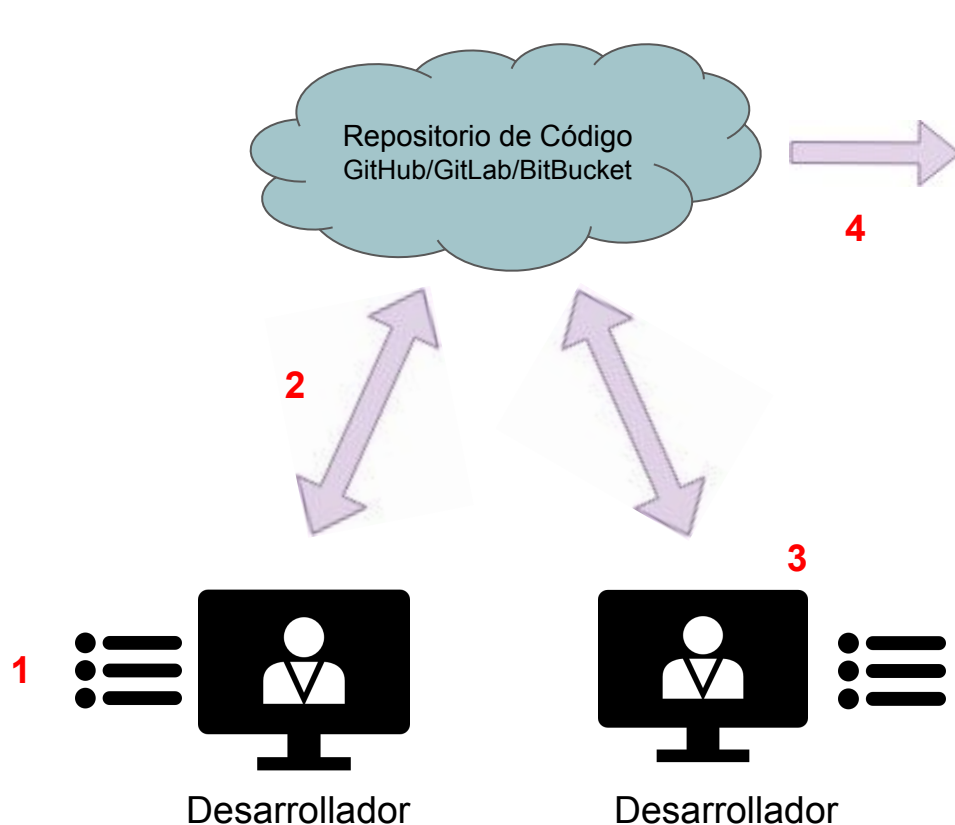
- Sistema versionado de código
- Distribuido → cada desarrollador tiene un repositorio local, el cual se sincroniza en un repositorio central
- Permite
 - Generar nuevas ramas de desarrollo
 - Incorporar a una rama cambios generados sobre ramificaciones previas

Localmente consta de

1. Un **repositorio local** (*directorio.git*)
 - a. Mantiene un historial de cambios en el código (commits, branches, etc).
2. Una **copia del código fuente** del repositorio.



Versionado del código con Git - Flujo de cambios típico



Flujo de trabajo con Git

1. Hacer cambios locales (commits)
2. Subir cambios (push)
3. Testear cambios (pull)
4. Actualizar "servidor"



Versionado del código - Git - Comandos

- **git clone** `<url_repo>`
 - clonar un nuevo repositorio p.e. `https://github.com/DSpace/DSpace.git`
- **git status**
 - ver el estado del repositorio (archivos nuevos, modificados, en conflicto)
- **git checkout -b** `<new_branch>`
 - crea una nueva rama y nos mueve a ella
- **git pull** [`<remote>` `<branch>`]
 - trae nuevos cambios a nuestro repositorio desde un servidor remoto.
 - por ejemplo: `git pull origin master`
- **git add** `<files>`
 - permite agregar archivos modificados a un commit
- **git commit**
 - crea un commit local con los archivos agregados mediante `git add`
- **git push** [`<remote>` `<branch>`]
 - lleva cambios locales a un servidor remoto

Ir a la [documentación oficial](#) de git para más información.



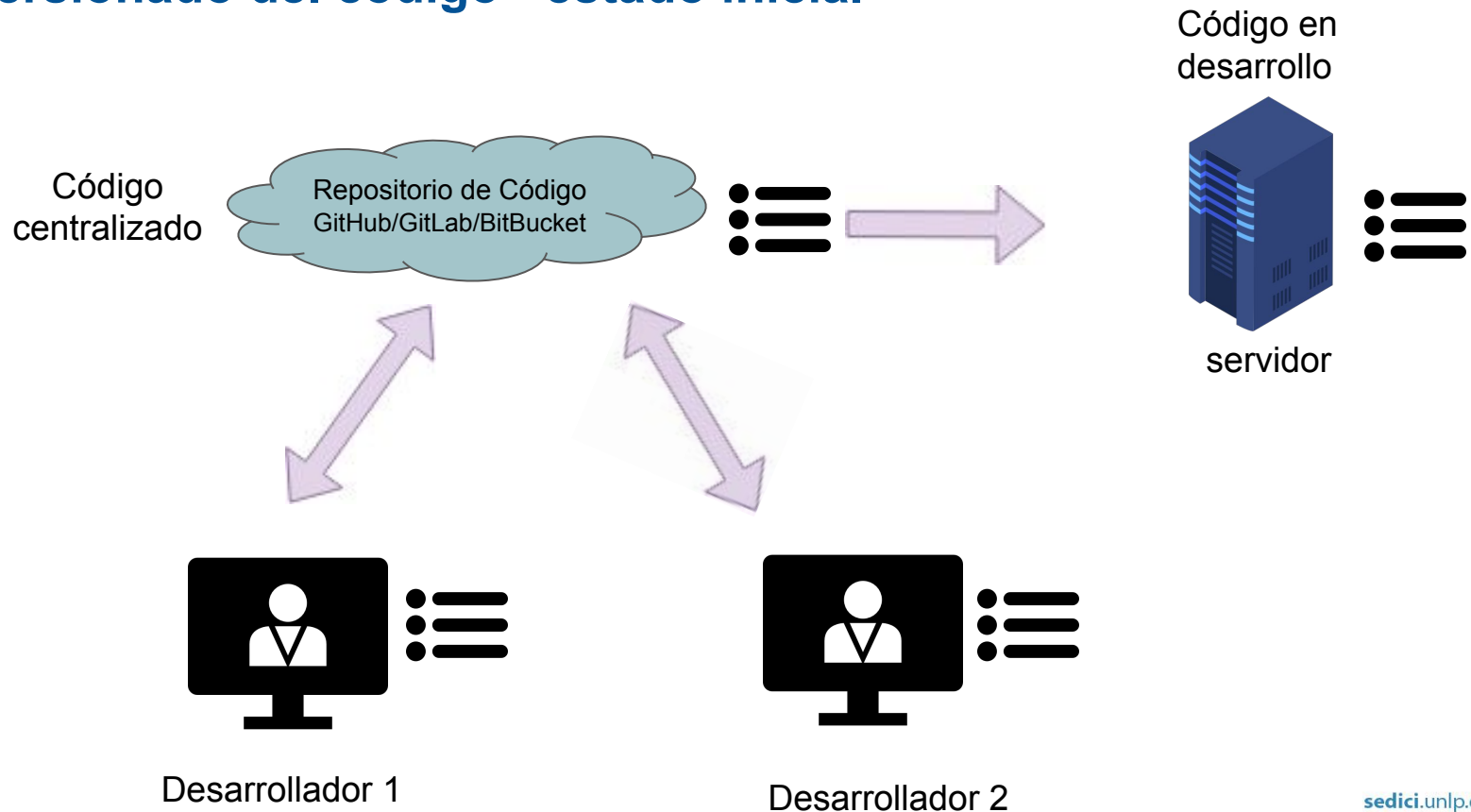
Circuito de desarrollo típico: caso simple

Circuito de desarrollo típico con desarrollo en **rama dev**

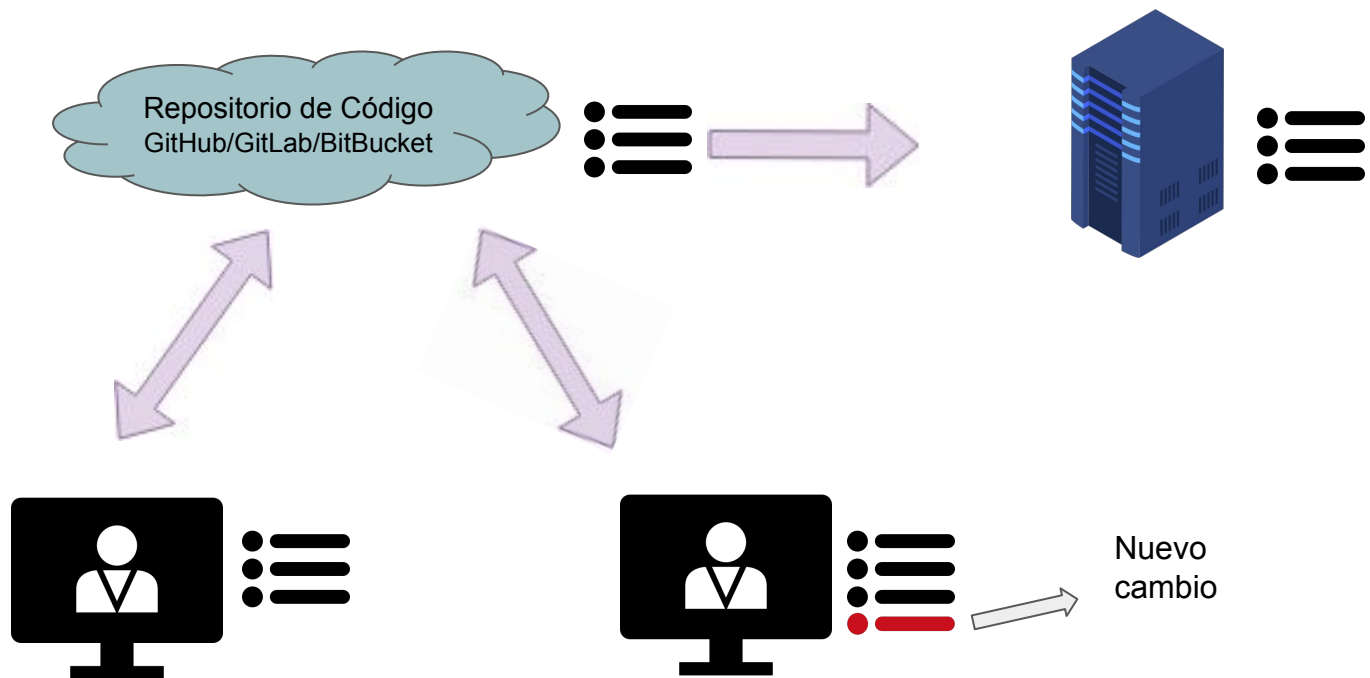
1. `git pull` → Traernos cambios pendientes
2. `git checkout dev` → trabajamos directamente en dev
3. Realizar los cambios → modificar el código con eclipse u otro editor
4. Compilar y actualizar el código
 - a. En el docker de DSpace → `docker-compose -f docker-compose.yml build`
5. Testear los cambios → desde la interfaz o directamente en el backend
6. `git add` y `git commit` → Agregar nuestros cambios al historial
7. `git push` → subir nuestros cambios
8. Creación del pull-request (de ser necesario)



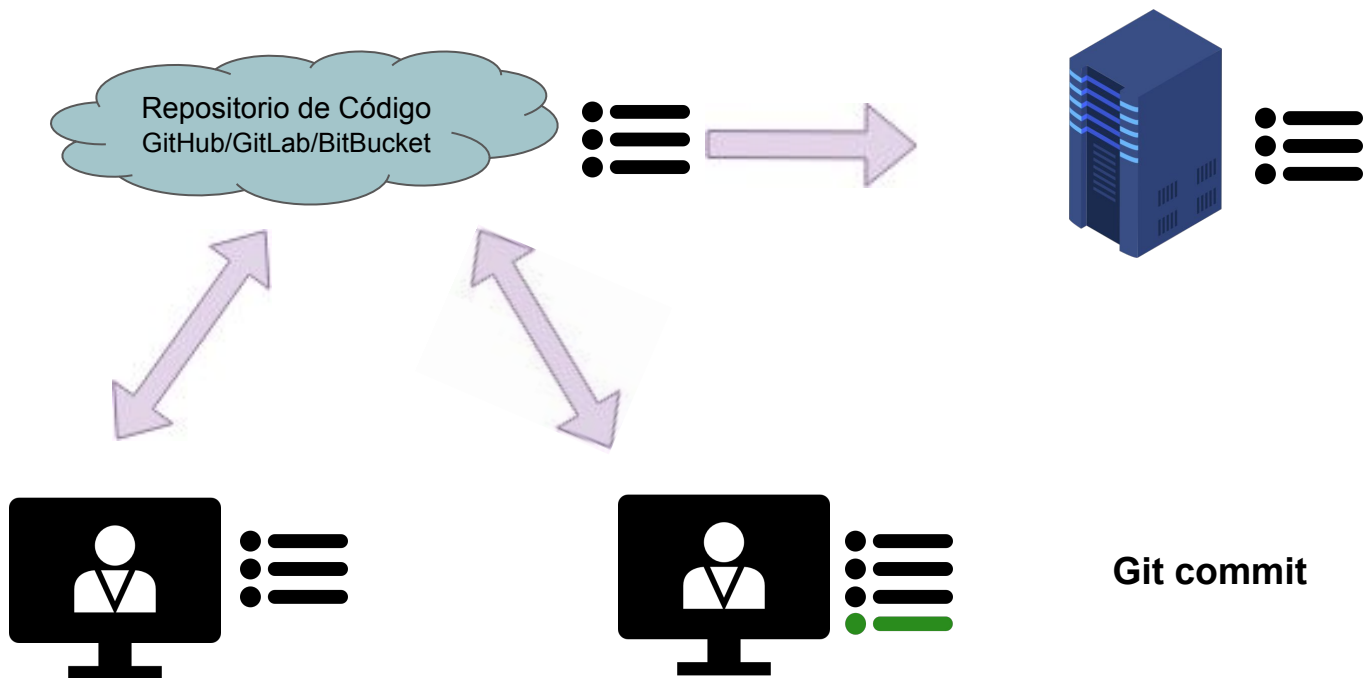
Versionado del código - estado inicial



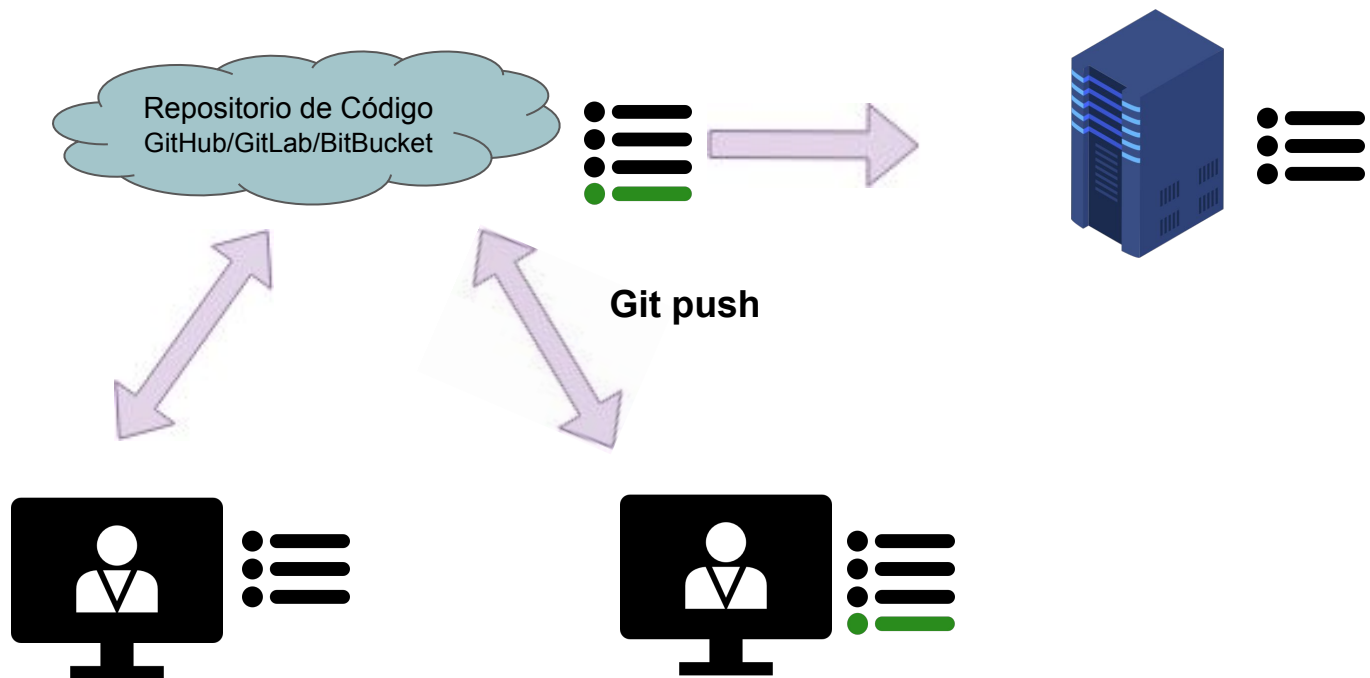
Versionado del código - cambio local



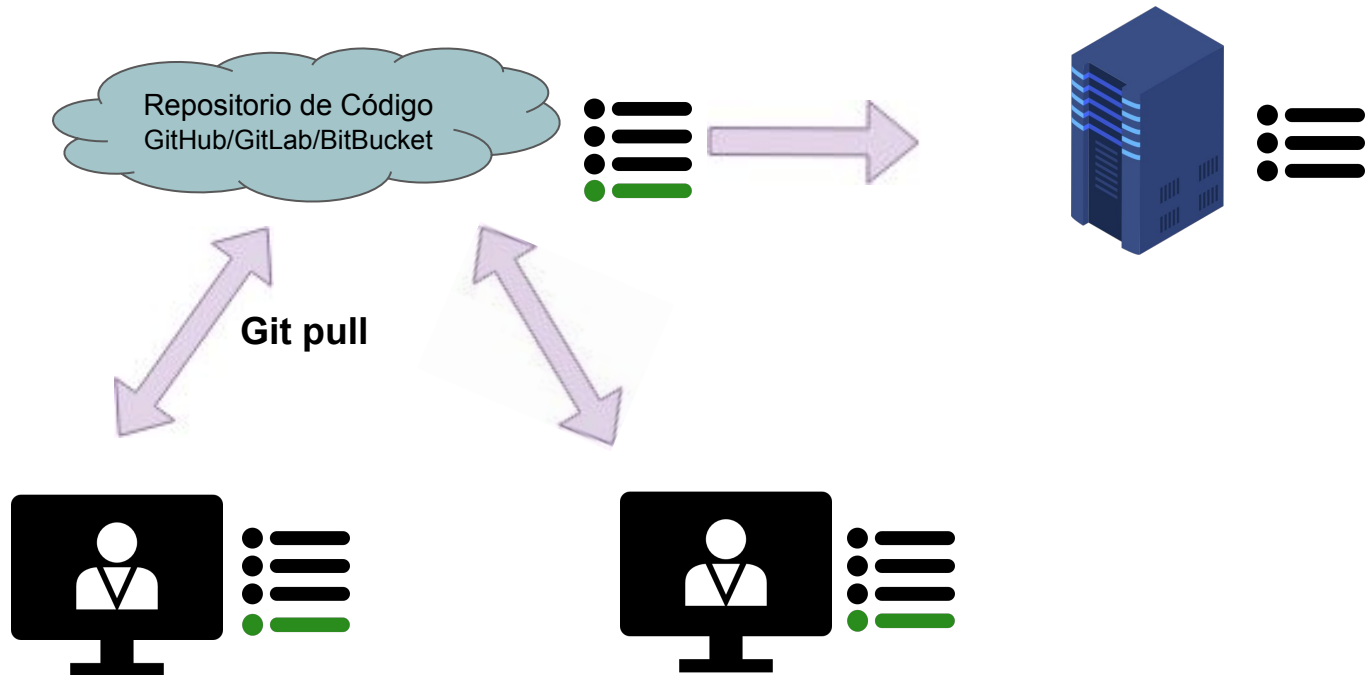
Versionado del código - cambio local versionado



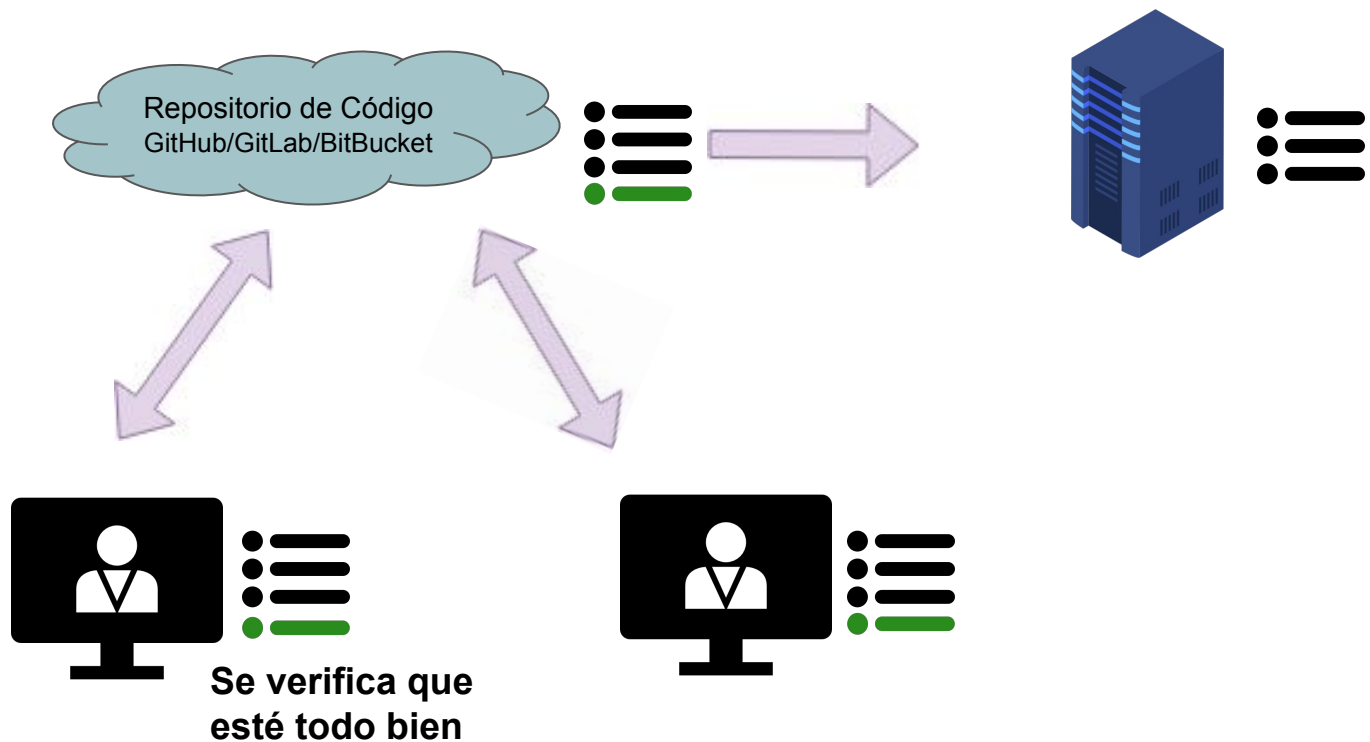
Versionado del código - cambio local compartido



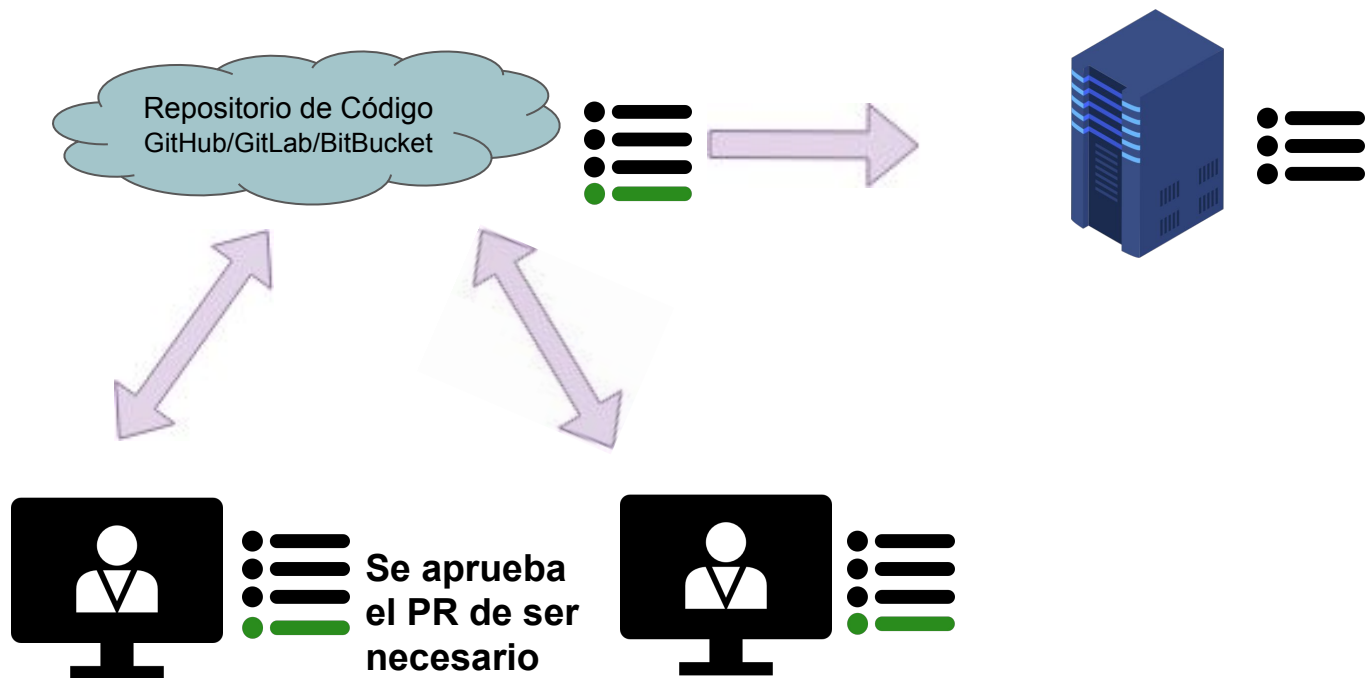
Versionado del código - cambio local - testing



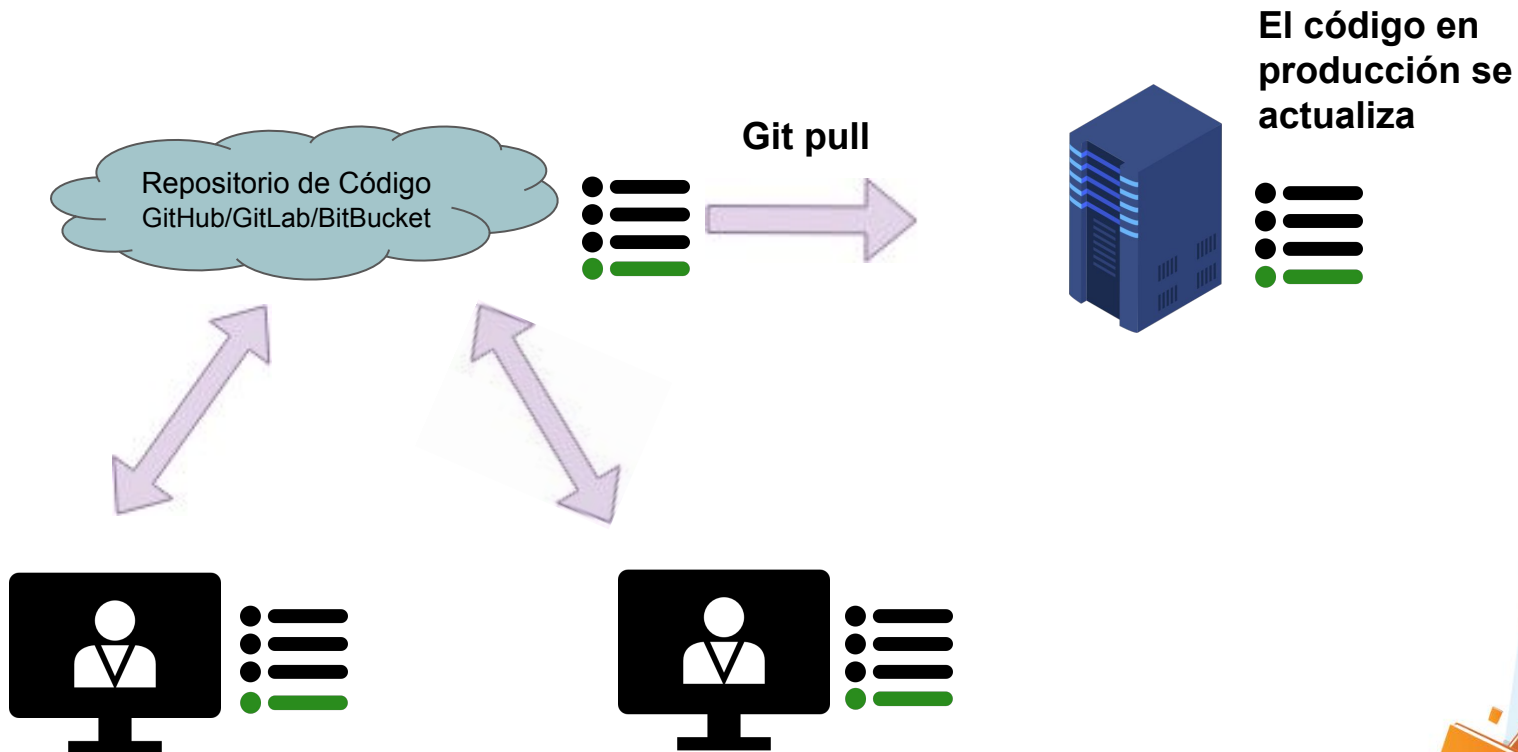
Versionado del código - cambio local - testing



Versionado del código - cambio local aprobado



Versionado del código - cambio local en producción



Github

- Es un servicio de alojamiento para repositorios *git*.
- Aloja repositorios públicos y privados.
- Permite hacer copias de otros repositorios (**forks**).
- Cada repositorio github tiene:
 - **repositorio git (para código)**
 - **issues**
 - **pull-requests**
 - **wikis**
 - **entre otros**



Git de Dspace/DSpace

Estructura de branches de dspace

- *main (Dspace 7)*
- *dspace-6_x*
- *dspace-5_x*
- *etc..*

Uso de topic branches

- *rest-demo*
- *preview*
- *etc..*

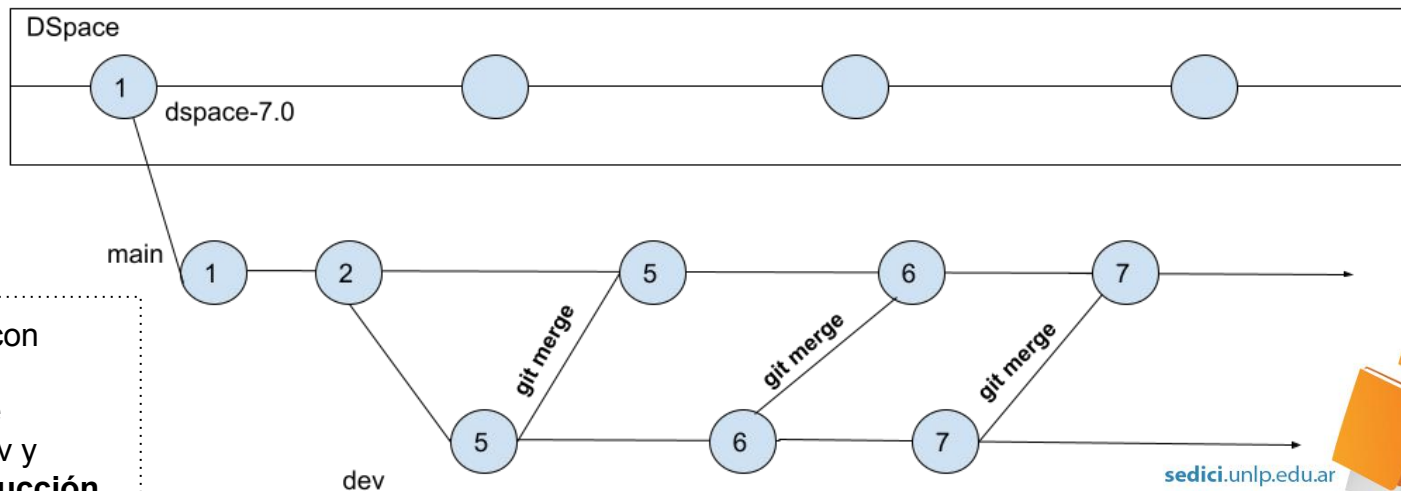
Tags (etiquetas para cada release) <https://github.com/DSpace/DSpace/tags>

- *dspace-7.0*
- *dspace-6.3*
- *dspace-5.10*



Git@DSpace - Prácticas recomendadas

- Al iniciar un nuevo proyecto Dspace
 - Crear nuevo repositorio a partir de un fork de Dspace/Dspace en GitHub
- Al menos dos ramas (branches)
 - Producción (main/master) → Cambios que están subidos al servidor
 - Desarrollo → Cambios que se incluirán en la siguiente release

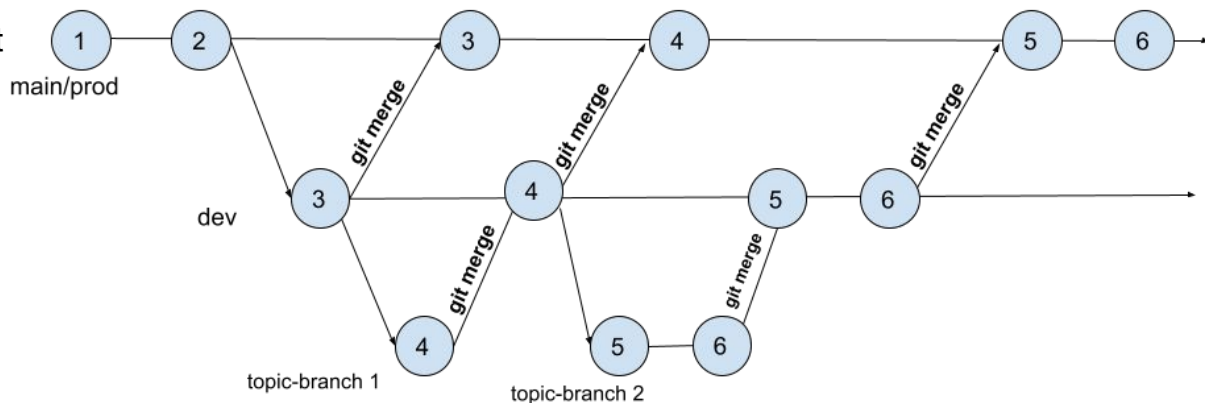


5, 6 y 7 son **commits** con cambios en código o configuraciones que se prueban primero en dev y luego se pasan a **producción**



Git@DSpace - Prácticas recomendadas

- Uso de topic branches
 - Ramas con nombres significativos en donde se trabaja en una característica específica del sistema
 - Nueva versión/release del proyecto
 - Nueva funcionalidad
 - Fix de un error
- Al sincronizar/unir dos ramas utilizar
 - Merge
 - Pull request



Git - Algunas buenas prácticas

1. No modificar a mano la instalación de producción
2. Actualizar primero desarrollo y testear los cambios
3. Desarrollar en una máquina local, preferiblemente en un entorno LOCAL
4. SIEMPRE hacer backups antes de actualizar el servidor
5. Antes de realizar cambios locales SIEMPRE hacer un git pull.
6. Nunca replicar a mano los cambios entre ramas, hacer git merge.



Entorno de desarrollo local - herramientas

Distintas herramientas para el desarrollo:

- Volúmenes de docker → persistir información más allá de la vida del container
 - Persistir la base de datos
 - Para que los cambios que realizamos en el código se reflejen en el container (bind mount)
- Editores de texto, IDEs → para modificar el código
 - Eclipse, Visual Studio, IntelliJ entre otros.
- Gitk → Es un visor gráfico de la historia del repositorio versionado.
 - Interfaz gráfica de git log, pero mostrando todas las ramas
- Meld → Herramienta que ayuda a comparar archivos, directorios y proyectos controlados por versiones.
 - Permite una visualización amigable de diff y sirve como ayuda para realizar merge entre ramas.



Personalizaciones en DSpace

Configuración básicas en BACKEND



Configuración de Dspace

1. [dspace]/config/local.cfg

- Configuración local, ajustes específicos, contraseñas, etc.
- No se versiona!

2. [dspace]/config/dspace.cfg

- Configuración principal y global del repositorio

3. [dspace]/config/modules/*.cfg

- Archivos de configuración de módulos

4. otras configuraciones adicionales [dspace]/config/*

- Archivos de configuración específicos de submission, workflow, xmlui, etc.

<https://wiki.lyrasis.org/display/DSDOC6x/Configuration+Reference>



Configuración - local.cfg

[dspace-source]/dspace/config/local.cfg

- Mantiene configuraciones propias de nuestra instalación que no sirven para el resto.
- Casos típicos
 - directorio de instalación de DSpace
 - configuración de email saliente
 - **código** de Google Analytics
 - usuario y **password** de acceso a base de datos
- Se pueden sobrescribir propiedades de otros módulos de configuración
 - datos de conexión con servidor LDAP
 - etc

[Como ver el valor actual de una propiedad de configuración](#)



Configuración - Prevalencia de local.cfg sobre el resto

local.cfg

```
dspace.dir = /var/repo/dspace/cic-digital/install  
dspace.baseUrl = http://digital.cic.gba.gob.ar  
oai.url = ${dspace.baseUrl}/oaipmh  
....
```

dspace.cfg

```
dspace.dir = /dspace  
dspace.baseUrl = http://localhost:8080  
...  
### i18n - Locales / Language ####  
default.locale = es  
webui.supported.locales = es, en, pt
```

modules/oai.cfg

```
oai.url = ${dspace.baseUrl}/oai  
...
```

Configuración resultante

```
dspace.dir = /var/repo/dspace/cic-digital/install  
dspace.baseUrl = http://digital.cic.gba.gob.ar  
  
oai.url = ${dspace.baseUrl}/oaipmh  
  
### i18n - Locales / Language ####  
default.locale = es  
webui.supported.locales = es, en, pt  
...
```



Configuración - dspace.cfg

[dspace-source]/dspace/config/dspace.cfg

- Reúne todas las configuraciones que controlan el funcionamiento del repositorio
- Define valores por default que pueden ser sobrescritos desde local.cfg
- Mantiene configuraciones comunes a todas las instalaciones del repo actual. Ej:
 - del módulo de exploración
 - idioma
 - plugins (autoridades, media-filters, authentication)
- ***define los archivos de configuración a cargar de otros módulos***
 - discovery.cfg
 - oai.cfg
 - etc.



Configuración - dspace.cfg

Permite definir configuraciones sobre:

- **Parámetros Generales**
 - `dspace.baseUrl` → hostname y puerto del servidor que utiliza dspace
 - `assetstore.dir` → ubicación del assetstore
- **Base de datos**
 - `db.name` → nombre de la base de datos
 - `db.url` → ip o host de la base de datos junto con el puerto
- **Emails SMTP**
 - `mail.server.disabled` → permite deshabilitar el envío de mails
 - `mail.server` → URL al servidor de mails

Ver en [dspace.cfg](#) los comentarios sobre las posibles configuraciones



Configuración - Módulos

[dspace]/config/modules/*.cfg

- Se encuentran archivos de configuración para los distintos módulos
 - rdf
 - solr
 - oai
 - tareas de curation
 - google analytics
 - authorities



Configuración - Archivos adicionales

Configuración adicional en [dspace]/config/:

- **default.license:**
 - licencia de depósito predeterminada utilizada por DSpace durante el proceso de envío
- **hibernate.cfg.xml :**
 - Configuración de la clase Hibernate para la base de datos de DSpace (casi nunca requiere cambio)
- **submission-forms.xml**
 - Configuración de formularios (en versiones anteriores input-forms.xml)
- **item-submission.xml:**
 - el proceso de envío predeterminado
- **launcher.xml :**
 - Configuración de la línea de comandos de DSpace ([dspace] / bin / dspace)
- **log4j.properties :**
 - Configuración de archivos de logs (generalmente ubicados en [dspace]/log)
- **workflow.xml (este se encuentra en [dspace]/config/spring/api) :**
 - Configuración del Workflow



Personalizaciones en DSpace

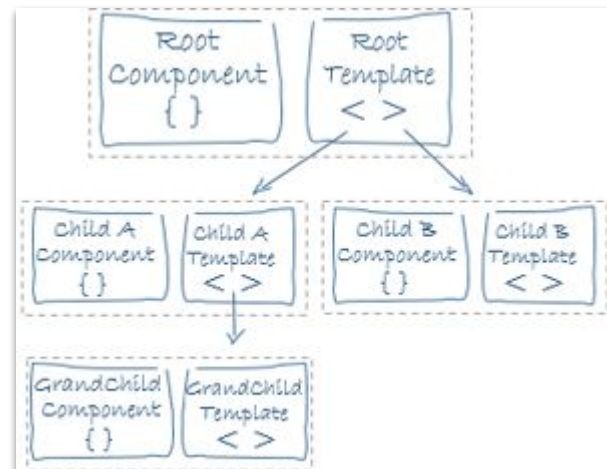
FRONTEND



Conceptos de Angular

Conceptos básicos de Angular.

- **component** → controla una parte de la pantalla (vista)
- **template** → plantilla HTML que define cómo renderizar un componente.
- **style** → estilos CSS/SCSS
- **theme** → conjunto de templates y estilos que componen el aspecto del sitio



Configuración de Angular

DSpace provee una forma de aplicar configuraciones propias

- Los archivos [config/config.*.yml](#)
 - No se versionan, similar a local.cfg del backend
- Se puede crear un archivo config.*.yml distinto para cada entorno
 - config.dev.yml → desarrollo
 - config.prod.yml → producción
 - config.yml → configuración por defecto, no debería modificarse
- Se puede definir:
 - Configuraciones del servidor
 - Conexión con la API-REST
 - Idiomas
 - Temas

```
rest:  
  ssl: false  
  host: localhost  
  port: 8080  
  namespace: /server  
  
# Submission settings  
submission:  
  typeBind:  
    # NOTE: which field to use wh
```



Personalización del tema

DSpace provee tres capas/directorios de temas por defecto:

- Base Theme
- DSpace Theme
- Custom Theme



Personalización del tema

- Base Theme
 - Dentro de **/src/app/**
 - Define el diseño principal de DSpace
 - Cada template se almacena en un subdirectorío con el nombre del componente Angular
- DSpace Theme
 - Dentro de **/src/themes/dspace**
 - Tema por defecto para DSpace 7.
 - Sólo proporciona personalizaciones que sobrescriben tema base.
 - Todas los templates se incluyen en el nivel del Tema Base



Personalización del tema

- Custom Theme
 - Dentro de **/src/themes/custom**
 - Este directorio permite crear un nuevo tema personalizado.
 - Componentes y templates vacíos que sobrescriben componentes base.

A través de la definición de nuestro propio **theme** basado en **Custom theme** se puede sobrescribir los estilos, plantillas, componentes y temas definidos por DSpace



Personalización del tema

Pasos para definir un tema propio

1. Definir y habilitar tema
2. Personalizar
3. Aplicar cambios



Personalización del tema - Definir el tema

1. Crear una copia de la carpeta **custom** en **src/themes**, y renombrarla con el nombre de nuestro tema.
2. En el archivo **angular.json**, en el arreglo “**styles**” agregar nuestro nuevo tema
 - El “**bundleName**” debe seguir el formato `${nombre-del-tema}-theme`.
 - Por ej, `iupa-theme` si el tema se llama `iupa`.
3. Habilitar el tema en los archivos **config.dev.yml** y **config.prod.yml**
 - Agregar nuestro tema al arreglo “**themes**”
 - Se puede definir también que nuestro tema aplique solo a ciertas partes del sistema definiendo un `handle/regex/uuid` para limitar la ruta en donde se aplica.

```
themes:  
  - name: 'cicba'  
    # Whenever this theme is activ  
    headTags:  
      # Insert <link rel="icon" hr  
      - tagName: link  
        attributes:  
          rel: icon  
          href: assets/cicba/image  
          type: image/x-icon
```



Personalización del tema - Personalizar el tema

4. Para realizar cambios generales en los estilos css, modificar el contenido de la carpeta “styles” dentro de nuestro tema.
5. Para incluir fuentes, imágenes o traducciones propias de nuestro tema, se deben agregar en el directorio “assets”
 - a. Por ejemplo, si se quiere agregar la imagen logo.png a nuestro tema:
src/themes/iupa/assets/images/logo.png
6. Para personalizar un componente específico, encontrar su archivo *.ts dentro de la carpeta del tema.
 - a. Para personalizar el estilo (css) del componente, descomentar la línea styleUrls comentada, y comente la activa. Entonces se va a usar el archivo *.scss de la misma carpeta del componente.
 - b. Para personalizar el html de tu componente, descomentar el templateUrl comentado, y comentar el activo. Entonces se va a usar el archivo *.html de la misma carpeta del componente.



Personalización del tema - Aplicar los cambios

7. Cualquier componente que no se quiera personalizar en el tema puede ser eliminado de la carpeta del tema
 - a. Esto reducirá ligeramente el tiempo de compilación y el tamaño de los paquetes
8. Recompilar la aplicación para ver los cambios.
 - a. **docker-compose -f docker/docker-compose.yml build** y **docker-compose -f docker/docker-compose.yml up**
 - b. O como alternativa **ng build** dentro del container



Internacionalización



Internacionalización

DSpace

- Están en un proyecto Git separado del proyecto central de DSpace
 - [dspace-api-lang](#)

DSpace Angular

- En la carpeta [assets/i18n](#) están los archivos con las traducciones de cada idioma.



Internacionalización - Frontend

Por defecto, DSpace utiliza el idioma del navegador del usuario

- Si tiene disponible el idioma del usuario, mostrará la interfaz en ese idioma.
- Si no, se mostrará por defecto en inglés o en otro idioma por defecto que se haya configurado.

Es posible agregar nuevos idiomas en el frontend Angular:

- En los archivos [config/config.\[dev|prod|test\].yml](#) se puede setear un arreglo de idiomas.
- Para agregar el español se debe agregar un nuevo elemento en ese arreglo
 - El "code" debe coincidir con el prefijo del nombre de un archivo de idioma.
 - El "label" es el texto que aparece en el selector de idioma de la UI.
 - "active" le permite decidir si ese idioma aparece en el selector de idioma o no.

```
defaultLanguage: en

languages:
  - code: en
    label: English
    active: true
  - code: de
    label: Deutsch
    active: true
  - code: es
    label: Español
    active: true
```



Internacionalización - Frontend

En la carpeta [assets/i18n](#) están los archivos con las traducciones de cada idioma.

- [es.json5](#) (español)
- [pt-BR.json5](#) (portugués de Brasil)
- [en.json5](#) (Inglés).

El prefijo antes de “.json5” (ej, pt-BR) debe coincidir con alguno de los códigos del arreglo de idiomas.

En [src/themes/\[NOMBRE_TEMA\]/assets/assets/i18n/*.json5](#) están los archivos con traducciones de cada tema en particular (en este caso el link es al custom theme).

- Para agregar nuevas traducciones se edita la key correspondiente en cada archivo de idioma al que se quiera traducir
- Es posible recibir parámetros entre `{{ }}`
- Por ej en es.json5:
- Para aplicar las traducciones del tema se debe ejecutar dentro del container:

```
"admin.registries.metadata.form.create": "Crear esquema de metadatos",  
"admin.registries.metadata.form.name": "Nombre",  
"admin.registries.bitstream-formats.edit.head": "Formato de Archivo: {{ format }}",
```

- `yarn merge-i18n -s src/themes/[theme-name]/assets/i18n`



Actividades

- Hacer el fork de DSpace a un repositorio propio
- Crear un nuevo subtema en el frontend
 - Agregar “Español” como idioma en el frontend y asignarlo como idioma default.
 - Cambiar el texto del Home

