

Configurando Angular no DSpace

Arturo Garduño-Magaña; Juan Manitta

Octubre 2025



Informações úteis antes de começar

Para tirar o máximo partido desta sessão, recomendamos:



A sessão será em espanhol com tradução simultânea para português.



Faça perguntas utilizando o botão de perguntas e respostas (não utilize o chat).



Consulte o Código de Conduta do Horário de Funcionamento da LA Referencia.



Esta sessão será gravada e partilhada juntamente com os diapositivos

Arturo Garduño-Magaña

Coordinador del proyecto, Lyrasis-LA Referencia

Juan Manitta

Desenvolvimento de software do projeto, Lyrasis-LA Referencia

Contente

- Apresentando a arquitetura Angular no DSpace 8.x/9.x
- Mostrar opções de configuração e personalização
- Compartilhar experiências com o nó de LA referência

De JSPUI/XMLUI para Angular

- DSpace 5/6 → 7 → 8/9
- De interfaces JSPUI/XMLUI ao Angular SPA
- Vantagens: modularidade, flexibilidade, consumo via API REST

De JSPUI/XMLUI para Angular

- **Caso 1:**

Migrar do **DSpace 6** ou **anterior**.

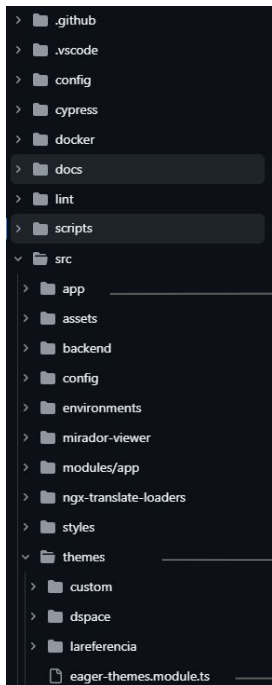
- **Caso 2:**

Migrar do **DSpace 7** ou **posterior**.

Do DSpace 6 ao DSpace 7+

- Branding e estilos (CSS, logotipos, etc.)
- Alterações de layout
- Recursos específicos
- Traduções

Estrutura do projeto



Pasta de componentes

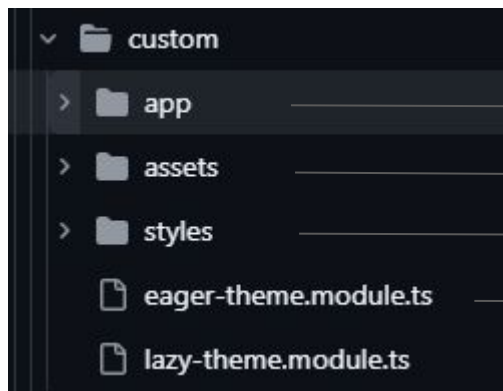
- header
- footer
- item

Pasta de tema

- Custom
- Dspace
- my-theme

Módulo para
activar el tema

Estrutura do projeto



Pasta de componentes do tema

Ativos (logotipos, imagens, etc.)

Arquivos de estilo, css, scss

Módulo para importar componentes para o tema

Criar e ativar tema

- Crie uma nova pasta dentro de "themes" (ou duplique a pasta personalizada e renomeie-a).
- Adicione esta nova pasta à sua configuração "angular.json".
- Ative este novo tema em seu config.*.yml.
- Ative o novo tema em "themes/eager-themes.module.ts".

Criar e ativar tema

Configuração **angular.json**

```
"styles": [  
  "src/styles/startup.scss",  
  {  
    "input": "src/styles/base-theme.scss",  
    "inject": false,  
    "bundleName": "base-theme"  
  },  
  {  
    "input": "src/themes/custom/styles/theme.scss",  
    "inject": false,  
    "bundleName": "custom-theme"  
  },  
  {  
    "input": "src/themes/dspace/styles/theme.scss",  
    "inject": false,  
    "bundleName": "dspace-theme"  
  },  
  {  
    "input": "src/themes/lareferencia/styles/theme.scss",  
    "inject": false,  
    "bundleName": "lareferencia-theme"  
  },  
  "node_modules/leaflet/dist/leaflet.css",  
  "node_modules/leaflet.markercluster/dist/MarkerCluster.Default.css",  
  "node_modules/leaflet.markercluster/dist/MarkerCluster.css"  
],
```

Nossa nova pasta

```
{  
  "input": "src/themes/lareferencia/styles/theme.scss",  
  "inject": false,  
  "bundleName": "lareferencia-theme"  
},
```

Criar e ativar tema

Configuração **config.*.yml**

```

- name: dspace
  headTags:
  - tagName: link
    attributes:
      rel: icon
      href: assets/dspace/images/favicons/favicon.ico
      sizes: any
  - tagName: link
    attributes:
      rel: icon
      href: assets/dspace/images/favicons/favicon.svg
      type: image/svg+xml
  - tagName: link
    attributes:
      rel: apple-touch-icon
      href: assets/dspace/images/favicons/apple-touch-icon.png
  - tagName: link
    attributes:
      rel: manifest
      href: assets/dspace/images/favicons/manifest.webmanifest
```

Substitua pelo nome do seu tema.

Outras personalizações são possíveis aqui, como favicon, ícone do Apple Touch, etc.

Você pode expandi-lo, adicionar vários temas, etc.

Criar e ativar tema

Temas de configuração/eager-themes.module.ts

```
import { NgModule } from '@angular/core';

// import { EagerThemeModule as DSpaceEagerThemeModule } from './dspace/eager-theme.module';
// import { EagerThemeModule as CustomEagerThemeModule } from './custom/eager-theme.module';
import { EagerThemeModule as LaReferenciaEagerThemeModule } from './lareferencia/eager-theme.module';

/**
 * This module bundles the eager theme modules for all available themes.
 * Eager modules contain components that are present on every page (to speed up initial loading)
 * and entry components (to ensure their decorators get picked up).
 *
 * Themes that aren't in use should not be imported here so they don't take up unnecessary space in the main bundle.
 */
@NgModule({
  imports: [
    // DSpaceEagerThemeModule,
    // CustomEagerThemeModule,
    LaReferenciaEagerThemeModule,
  ],
})
export class EagerThemesModule {
}
```

Importe nosso módulo do novo tema

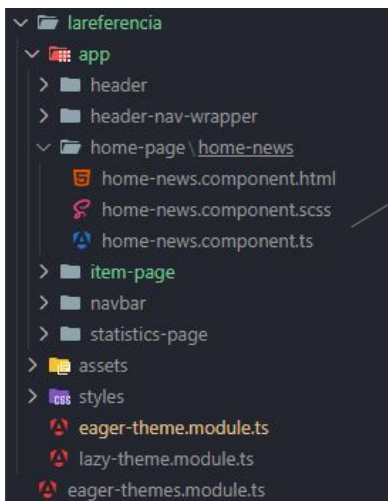
```
@NgModule({
  imports: [
    // DSpaceEagerThemeModule,
    // CustomEagerThemeModule,
    LaReferenciaEagerThemeModule,
  ],
})
```

Personalizar componentes

- **Copie a pasta do componente** para o nosso tema personalizado (se possível, de "custom").
- **Declare o componente** em `edger-theme.module.ts` (no nosso tema).
- Modifique a **importação do HTML, CSS, etc.** (o que você quiser personalizar) no `.ts` do nosso componente.
-

Personalizar componentes

Copie a **pasta do componente**



Componentes para personalizar

- Pastas e subpastas.
- Usaremos arquivos .html, .css, .ts e de teste.

Personalizar componentes

Modifique a importação de html, css etc.

```
import { Component } from '@angular/core';

import { HomeNewsComponent as BaseComponent } from '../../../../../app/home-page/home-news/home-news.component';

@Component({
  selector: 'ds-themed-home-news',
  styleUrls: ['../../../../app/home-page/home-news/home-news.component.scss'],
  templateUrl: './home-news.component.html',
  standalone: true,
})

/**
 * Component to render the news section on the home page
 */
export class HomeNewsComponent extends BaseComponent {}
```

Importando o .scss
padrão do tema base

Importando .html
personalizado (pasta do
tema)

IMPORTANTE: Estende-se do
componente original do tema base do
DSpace, renomeado como
"BaseComponent"

Personalizar componentes

Declare o componente em eager-theme.module.ts

```
const DECLARATIONS = [  
  ... ENTRY_COMPONENTS,  
  HomeNewsComponent,  
  HeaderComponent,  
  HeaderNavbarWrapperComponent,  
  NavbarComponent,  
  PublicationComponent,  
];  
  
@NgModule({  
  imports: [  
    CommonModule,  
    RootModule,  
    ... DECLARATIONS,  
  ],  
  providers: [  
    ... ENTRY_COMPONENTS.map((component) => ({ provide: component })),  
  ],  
})
```

Declare o(s) componente(s) que
queremos personalizar

Atenção! Importar o componente de rota
do nosso

```
import { RootModule } from '../..//app/root.module';  
import { HeaderComponent } from './app/header/header.component';  
import { HeaderNavbarWrapperComponent } from './app/header-nav-wrapper/header-nav-wrapper.component';  
import { HomeNewsComponent } from './app/home-page/home-news/home-news.component';  
import { NavbarComponent } from './app/navbar/navbar.component';  
import { PublicationComponent } from './app/item-page/simple/item-types/publication/publication.component';
```

Criar novos componentes

- Crie uma nova pasta em nosso tema (com o nome do componente)
- Crie os arquivos necessários (.ts, .html, .scss)
- Declare o componente em nosso eager-theme.module.ts
- Importe o componente autônomo onde queremos usá-lo.

```
@Component({
  selector: 'ds-themed-header-navbar-wrapper',
  styleUrls: ['header-navbar-wrapper.component.scss'],
  templateUrl: 'header-navbar-wrapper.component.html',
  standalone: true,
  imports: [
    AsyncPipe,
    ThemedHeaderComponent,
    ThemedNavbarComponent,
    TranslateModule,
  ],
  animations: [slideMobileNav],
})
```

```
imports: [
  AsyncPipe,
  ThemedHeaderComponent,
  ThemedNavbarComponent,
  TranslateModule,
],
```

Serviços e solicitações http

- Em nosso precisamos ter aplicativo/núcleo/dados
- Crie nosso arquivo service.ts com a lógica do nosso serviço
- Consuma nosso serviço a partir de nossos componentes.

Personalizar estilos

Global

Localize a pasta de estilos dentro do nosso tema

Edite como preferir:

- `_estilos globais.scss`
- `_theme_css_variable_overrides.scss`
- `_theme_sass_variable_overrides.scss`

Local

- Localize e edite o arquivo `.scss` na pasta do componente para personalizar

Personalizar estilos

Globales - bootstrap (_theme__sass_variable_overrides.scss)

```
// Override semantic colors here
$primary:    #92c642; // Gray
$secondary:  #495857; // As Bootstrap $gray-700
$success:    #92c642; // Lime
$info:       #1e6f90; // Light blue
$warning:    #ec9433; // Orange
$danger:     #cf4444; // Red
$light:      #f8f9fa; // As Bootstrap $gray-100
$dark:       #43515f; // Dark blue
```

É possível personalizar com as cores da instituição.

É possível sobrescrever outras variáveis Bootstrap existentes neste arquivo.

Personalizar estilos

Globales - DSpace (_theme_css_variable_overrides.scss)

```
:root {  
  
  @include media-breakpoint-up(md) {  
    --ds-header-logo-height: 40px;  
    --ds-header-height: 80px;  
  }  
  @include media-breakpoint-down(sm) {  
    --ds-header-logo-height: 50px;  
    --ds-header-height: 90px;  
  }  
  
  --ds-banner-text-background: rgba(0, 0, 0, 0.58);  
  --ds-banner-background-gradient-width: 300px;  
}
```

É possível personalizar variáveis que o DSpace usa por padrão

É possível adicionar nossas próprias variáveis (evitando usar --ds ou --bs)

Personalizar estilos

Globales (_global-styles.scss)

```
// imports the base global style
@import '../.../styles/_global-styles.scss';

.facet-filter, .setting-option, .advanced-search {
  background-color: var(--bs-light);
  border-radius: var(--bs-border-radius);
}

&.p-3 {
  // Needs !important because the original bootstrap class uses it
  padding-top: 0.5rem !important;
  padding-bottom: 0.5rem !important;
}

.bg-secondary {
  background-color: var(--bs-primary);
}

h4, .h4 {
  font-size: 1.1rem
}
```

É possível editar classes globais usadas em nosso tema.

IMPORTANTE: Na maioria das vezes é mais rápido e fácil usar os dois métodos mencionados acima

Traduções (i18n)

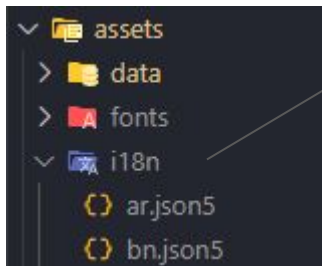
Alterações no núcleo (simples mas não recomendadas)

- Localize `src/assets/i18n/**/*.json5`
- Localize o valor que queremos alterar, por exemplo “`menu.section.browse_global`”

Mudanças no nosso tópico

- Criar pasta “i18n” em `src/themes/[my-theme]/assets`
- Crie o arquivo para traduzir, por exemplo, “`is.json5`”
- Adicione apenas a linha que queremos traduzir.
- Quando terminar, execute o comando “`npm merge-i18n -- -s src/themes/[theme-name]/assets/i18n`”

Traduções (i18n)



Pasta i18n e arquivos .json5 com traduções

Chave para traduzir

Valor a traduzir

```
// "menu.section.browse_community_by_issue_date": "By Issue Date",  
"menu.section.browse_community_by_issue_date": "Por fecha de publicación",
```

Exemplos de personalização



CONSEJO
NACIONAL
DE RECTORES

Repositorio Institucional



CONSEJO
NACIONAL
DE RECTORES

Repositorio Institucional

Comunidades Todo DSpace Estadísticas

🔍 Iniciar sesión

Buscar en el repositorio ...

🔍 Buscar

¿Qué es el repositorio CONARE?

El Repositorio Institucional es la herramienta que el CONARE pone a disposición de la sociedad para facilitar el acceso a los resultados de la investigación, extensión y acción social realizada por Programas, Proyectos, Comisiones y Equipos de Trabajo. Asimismo, aumentar la visibilidad de la producción intelectual institucional y contribuir a la preservación de los documentos digitales allí depositados.

Elija una para listar sus colecciones:



CeNAT
Centro Nacional de Alta Tecnología



OPES
Oficina de Planificación
de la Educación Superior



PEN PROGRAMA
ESTADO DE LA NACIÓN

Comunidades Todo DSpace Estadísticas

🔍 Iniciar sesión

Inicio - Oficina de Planificación de...



OPES
Oficina de Planificación
de la Educación Superior

URL permanente para esta comunidad: <https://hdl.handle.net/20.500.12337/2>

Examinar

Subcomunidades y colecciones Por fecha de publicación Por autor/a Por título Por materia Por serie

Subcomunidades de esta comunidad

Mostrando 1 - 1 de 1



ARCHIVO VERTICAL

Subcomunidad privada para uso interno de la Biblioteca del CONARE

Colecciones de esta comunidad

Mostrando 1 - 9 de 9



Cores, imagens, logotipos

Cabeçalho, notícias domésticas,
comunidades, barra de navegação,
etc

Atualização da interface do usuário

- Verifique os requisitos de versão, Node, Yarn, PM2 etc.
- Faça backup de nossa pasta em “temas” e arquivos config.yml
- Confirme as alterações pendentes na filial
- Faça uma mesclagem para a próxima versão
- Possível conflito “package.json”, aceitar versão “entrada”
- Possível conflito “package-lock.json ou yard”, aceitar “entrada” ou excluir e regenerar
- Manual de mesclagem de possível conflito “config.yml”, novas opções de configuração
- Instale dependências (yarn ou npm), primeiro exclua node_modules

Atualização da interface do usuário

- Valide nosso tópico. Esta pasta não foi substituída porque não existe no código-fonte do DSpace.
 - Levante o projeto em desenvolvimento e verifique se as personalizações estão corretas, pois nomes de variáveis, nomes de classes ou componentes que alteram nossa personalização podem ter mudado no código-fonte.
 - Se houver erros, pesquise e resolva-os no desenvolvimento um por um

Caso particular 7 -> 8+

DSpace 7 x usou Angular 14 e a arquitetura **NgModules**.

O Dspace 8 em diante saltou para o Angular 16+, adotando a arquitetura **Standalone Components**

- Converta todos os componentes do nosso tema personalizado para Standalone

Caso particular 7 -> 8+

Dspace 7x

```
/**
 * Component representing the public navbar
 */
@Component({
  selector: 'ds-navbar',
  styleUrls: ['./navbar.component.scss'],
  templateUrl: './navbar.component.html',
  animations: [slideMobileNav]
})
export class NavbarComponent extends BaseComponent {
}
```

NOVO: seletor “independente”

Agora as dependências e importação de outros

os componentes são colocados dentro do componente

Dspace 8+

```
/**
 * Component representing the public navbar
 */
@Component({
  selector: 'ds-themed-navbar',
  styleUrls: ['./navbar.component.scss'],
  templateUrl: './navbar.component.html',
  animations: [slideMobileNav],
  standalone: true,
  imports: [
    AsyncPipe,
    NgbDropdownModule,
    NgClass,
    NgComponentOutlet,
    ThemedUserMenuComponent,
    TranslateModule,
  ],
})
export class NavbarComponent extends BaseComponent {
}
```

Boas práticas

- Configurações de personalização separadas:
 - Use a configuração o máximo possível, mantenha as personalizações no mínimo Alterações de
- Documentos no Git:
 - Use bons commits para atualizações futuras
- Use ambientes de teste:
 - Nunca atualize e execute a mesclagem na produção
- Não modifique o núcleo Angular:
 - Para evitar perdas em atualizações e/ou conflitos massivos de mesclagem

Problemas e soluções comuns

Erros de cache angular

- **Problema:** Os usuários ainda veem a versão antiga do site (por exemplo, banner antigo) após uma atualização.
- **Causa principal:** Não é o cache do navegador, mas o DSpace "Service Worker" (PWA) que salva agressivamente arquivos antigos.
- **Correção rápida:** forçar recarga no DevTools (F12) > guia "Aplicativo" > "Trabalhadores de serviço" > ativar "Atualização na recarga".

Conflitos com dependências

- **Problema:** yarn install ou npm install (dependendo da versão) falha com erros de dependência de pares após mesclar uma nova versão do DSpace.
- **Solução rápida:** exclua as pastas node_modules e o arquivo yard.lock antes de reinstalar e sempre use yard ou npm (conforme apropriado).

Problemas de versão

- **Problema:** O frontend carrega, mas não exibe dados, com erros 404 ou 500 em chamadas de API (por exemplo, ao efetuar login).
- **Causa principal:** A versão frontend (Angular) não corresponde à versão backend (API REST). Eles devem ser idênticos (por exemplo, DSpace 8.1 com DSpace 8.1).

Envolvimento comunitário

Acesse

www.menti.com

Digite o código

6847 8200

Ou use seu celular e
acesse via código QR



Perguntas e respostas

¡Obrigado pela sua ajuda!

Para qualquer dúvida, escreva para:

Arturo Garduño-Magaña
arturo.garduno.magana@gmail.com