

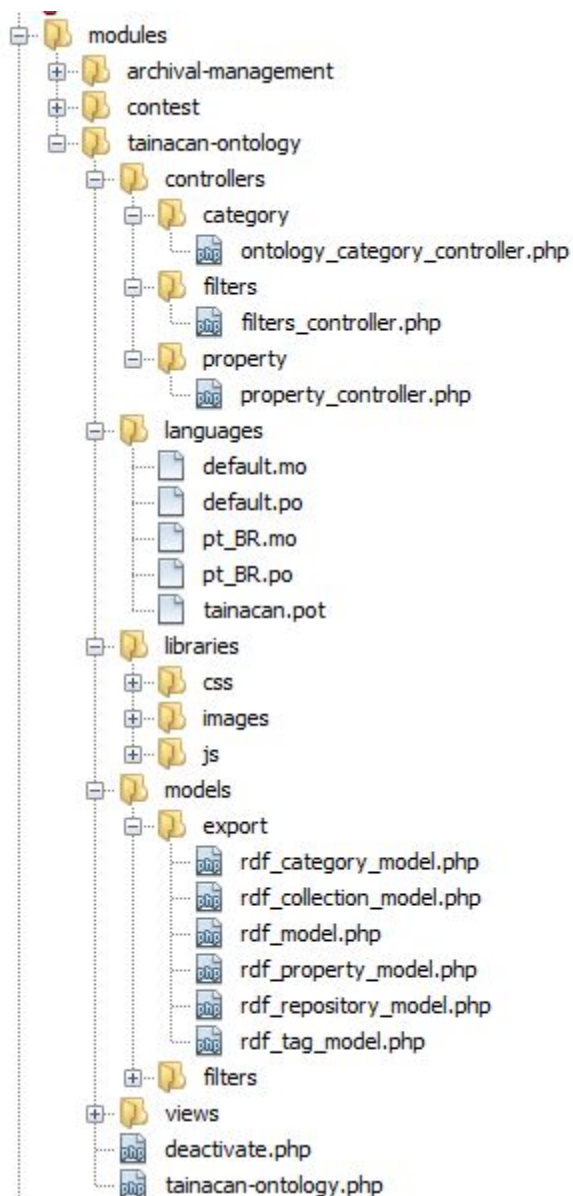
Módulos Tainacan

A arquitetura de módulos foi criada com o objetivo de permitir derivações do Tainacan sem a necessidade de alterações em seu código 'core'.

Esta arquitetura utiliza-se da estrutura de actions e filters do próprio Wordpress, permitindo a criação e a modificação de diversos comportamentos do Tainacan. Em anexo está uma lista com todos os actions e filters já cadastrados na plataforma. Esta listagem está em constante mudança, pois sempre está surgindo novas necessidades e novas funcionalidades no próprio Tainacan 'core'.

Estrutura

Os arquivos de um módulo ficam dentro da pasta *modules* do tema. Sua arquitetura assemelha-se com o próprio Tainacan, separando em pastas as camadas MVC, os arquivos de línguas e bibliotecas utilizadas pelo módulo



É obrigatório a criação de um arquivo com o mesmo nome do módulo, este será o arquivo 'index', onde é iniciada a execução dos todos scripts, abaixo é mostrado um parte do código fonte de um arquivo 'inicializador' do módulo de ontologia.

```
* #14 - ALTERACAO DO FORMULARIO DE ADICAO DE PROPRIEDADE DE OBJETO
* #15 - ALTERACAO DA PAGINA DO ITEM PARA PROPRIEDADE DE DADOS
* #16 - ADICIONA O BOTAO DE EDITAR PROPRIEDADE NA PAGINA DA PROPRIEDADE
* #17 - ADICIONA NO MENU DA COLECAO A OPCAO DE FILTROS
* #18 - ALTERA O THUMBNAIL DOS ITEMS/COLECAO
*
* @author: EDUARDO HUMBERTO
*/

define('MODULE_ONTOLOGY', 'tainacan-ontology');
define('ONTOLOGY_CONTROLLERS', get_template_directory_uri() . '/modules/' . MODULE_ONTOLOGY);
load_theme_textdomain("tainacan", dirname(__FILE__) . "/languages");

##### #1 ADICIONANDO OS SCRIPTS DESTE MODULO #####
add_action('wp_enqueue_scripts', 'tainacan_ontology_js');
function tainacan_ontology_js() {
    wp_register_script('tainacan-ontology',
        get_template_directory_uri() . '/modules/' . MODULE_ONTOLOGY . '/lib',
        $js_files = ['tainacan-ontology'];
    foreach ($js_files as $js_file):
        wp_enqueue_script($js_file);
    endforeach;
}

add_action('wp_enqueue_scripts', 'tainacan_ontology_css');
function tainacan_ontology_css() {
    $registered_css = [
        'tainacan-ontology' => '/libraries/css/tainacan-ontology.css'
    ];
    foreach ($registered_css as $css_file => $css_path) {
        wp_register_style($css_file, get_template_directory_uri() . '/modules/' . $css_path);
        wp_enqueue_style($css_file);
    }
}

#####
```

Apenas neste trecho, é possível notar a inserção da localização da pasta dos arquivos de línguas. Estes arquivos sobrepõem os do tema, por isso as strings traduzidas neste local reflete para todo o Tainacan. Nota-se, também o carregamento dos scripts do módulo, onde utiliza-se os mesmos processos do wordpress padrão.

O arquivo *deactivate.php* não é obrigatório, ele é executado se cadastrado no arquivo de ativação de módulos (mostrado na próxima seção), sendo útil para remoção de dados cadastrados no banco de dados que não serão utilizados em outro módulo ou na versão default do Tainacan.

Ativação

Apenas a inserção dos arquivos de um módulo não o ativa automaticamente, no Tainacan dentro da pasta *config* no arquivo *config.php*, precisa ser alterado para reconhecer um novo módulo. Abaixo temos o arquivo *config.php*:

```
<?php
2
3 /*
4  *
5  * Arquivo de Configuracao do modo de operacao do Tainacan
6  *
7  */
8
9 /*
10 # modo = 0; 0 ou nulo é o funcionamento atual padrão
11
12 # modo = 1; gestão arquivística e museológica
13
14 # modo = 2; gestão de ontologias
15
16 # modo = 3; Modo de debates Tainacan
17 */
18
19
20
21 global $config;
22 $config['mode'] = 0;
23 // o nome deve ser o mesmo da pasta e também o nome do arquivo aonde esta
24 // o carregamento do modulo
25 $config['active_modules'] = []; //['archival-management']['contest']['tainacan-o
26 //importante para a retirada de metadados criados
27 $config['deactive_modules'] = [];
28 //// ['tainacan-ontology']; //['archival-management']
29 // os metadados a serem inseridos como default
```

Desta forma em que o arquivo se encontra, não está ativado nenhum módulo. Para cada modo de operação deve ser alterado a variável `$config['mode']` e então deve ser colocado dentro do array `$config['active_modules']` os módulos para este modo de operação(Nota: o nome do módulo deve ser exatamente igual ao nome da pasta em que se encontra), desta forma é permitido a utilização de vários módulos para diferentes modos de operação.

Anexo - Actions e Filters Tainacan

1. Thumbnail Itens/coleções

`apply_filters(alter_thumbnail_items, $icon)`

list.php (object)

retorna o icone a ser usado na listagem de itens

`apply_filters(alter_thumbnail_collections, $icon)`

list.php (object)

retorna o icone a ser usado na listagem de coleções

2. Adição de metas padrões

`do_action('add_new metas_category', $category_root_term);`

functions.php - create_category_terms()

Adiciona metadados no repositório assim se estende por todas coleções

`do_action(add_new metas_property, $category_root_term);`

functions.php - create_property_terms()

Adiciona metadados na propriedade raiz

`do_action(add_new metas_property_data, $category_root_term);`

functions.php - create_property_terms()

Adiciona metadados na propriedade de dados raiz

do_action(add_new metas_property_object,\$category_root_term);

functions.php - create_property_terms()

Adiciona metadados na propriedade de objetos raiz

do_action(add_new metas_event_create_category,\$event_create_term);

functions.php - create_event_terms()

Adiciona metadados no evento de criação de categoria

do_action(add_new metas_event_property_data,\$event_term);

functions.php - create_event_terms()

Adiciona metadados NOS EVENTOS de propriedade de dados

do_action(add_new metas_event_property_object,\$event_term);

functions.php - create_event_terms()

Adiciona metadados NOS EVENTOS de propriedade de objeto

3.Design/Diversos

do_action(add_button_edit_property)

page.php (propriedades)

permite adicionar botao de editar para propriedade em sua page de descricao

```
do_action(insert_new_contextmenu_dynatree)
```

single.php

permite adicionar um novo context menu para um dado dynatree

```
do_action('add_configuration_menu_tainacan');
```

header_collection.php

esta acao permite a inserção de itens no menu da colecao abaixo do item de categorias

```
apply_filters( 'show_custom_add_item_button', '' )
```

single.php

este filtro retorna um html para ser apresentado no lugar do botao padrao

```
do_action( 'before_tree' )
```

menu_left.php

acao que insere o botao para abrir o mal para inserção de facetas

```
do_action('item_type_attributes')
```

create.php/edit.php

ação que permite a criação de atributos para o campo tipo do item, podendo assim
lhe-esconder

```
do_action('item_from_attributes')
```

create.php/edit.php

ação que permite a criação de atributos para o campo Origem do item

`do_action('item_content_attributes')`

`create.php/edit.php`

ação que permite a criação de atributos para o campo content do item

`do_action('item_thumbnail_attributes')`

`create.php/edit.php`

ação que permite a criação de atributos para o campo thumbnail do item

`do_action('item_source_attributes')`

`create.php/edit.php`

acao que permite a criacao de atributos para o campo source do item

`do_action('item_tags_attributes')`

`create.php/edit.php`

acao que permite a criacao de atributos para o campo tags do item

`do_action('item_property_term_attributes')`

`create.php/edit.php`

acao que permite a criacao de atributos para o campo property terms do item

`do_action('item_attachments_attributes')`

`create.php/edit.php`

acao que permite a criacao de atributos para o campo source do item

`apply_filters('collection_object', $collection_object)`

`collection_model.php`

este filtro recebe o nome do objeto da coleção permitindo assim sua alteração para uso na categoria raiz

4.Menu da coleção

`do_action(menu_collection_search_configuration)`

`header_collection.php`

ação que permite a criação de atributos html para o menu search da coleção

`do_action(menu_collection_property_and_filters_configuration)`

`header_collection.php`

ação que permite a criação de atributos html para o menu property and filters da coleção

`do_action(menu_collection_property_configuration)`

`header_collection.php`

ação que permite a criação de atributos html para o menu property da coleção

`do_action('menu_collection_social_configuration')`

`header_collection.php`

ação que permite a criação de atributos html para o menu social_configuration da coleção

`do_action('menu_collection_license')`

header_collection.php

ação que permite a criação de atributos html para o menu Licenças da coleção

`do_action('menu_collection_import')`

header_collection.php

ação que permite a criação de atributos html para o menu Importar da coleção

`do_action('menu_collection_export')`

header_collection.php

ação que permite a criação de atributos html para o menu no dropdown Exportar da coleção

`do_action('menu_repository_social_api')`

header.php

ação que permite a criação de atributos html para o menu no dropdown Social api do repositório

`do_action('menu_repository_license')`

header.php

ação que permite a criação de atributos html para o menu no dropdown Licenças do repositório

`do_action('label_add_item')`

create.php

esta ação que gera um label personalizado para o título Adição de novos itens

`do_action('label_edit_item')`

edit.php

esta ação gera um label personalizado para o título Edição de itens

`do_action('optional_message')`

create.php

mensagem opcional para ser usado nos diversos campos do tainacan

`apply_filters(show_category_root_in_edit_category, $is_showed)`

visualization_model.php

este filtro determina se sera mostrado a categoria raiz na Edição de uma categoria

5.FILTROS

`do_action(add_standart_options_label_filters)`

edit.php (Filters)

adiciona labels para o selectbox de facetas padrões

`do_action(add_optiongroup_label_filters)`

edit.php (Filters)

permite adicionar um option group para o selectbox de facetas padrões

```
do_action(add_widget,$facet)
```

edit.php (Filters)

adciona o widget que sera mostrado na home da colecao

```
apply_filters( add_widgets_filters, $property_id)
```

search_model.php

este filtro recebe o id da propriedade/filtro para retornar as opções de visualização na Edição de filtros deve retornar neste formato

```
$var['range'] = __('Range','tainacan');  
$var['from_to'] = __('From/To','tainacan');
```

```
apply_filters( add_widgets_filters_in_view, $property_id)
```

visualization_model.php

Diferentemente da anterior este filtro busca dados para serem mostrados na view menu left

este filtro recebe o id da propriedade/filtro para retornar os valores

```
‘  
    $var[id] = $property_id;  
    $var[name] = __('From/To','tainacan');
```

```
apply_filters( get_filter_name, $property_id)
```

search_model.php

retorna o nome a ser usado na view de acordo o id cadastrado no selectbox

6.Modal Categorias

`do_action(add_option_in_add_category)`

single.php

Adiciona um label no header do modal de categoria

`do_action(show_option_in_add_category)`

single.php

permite adicionar um form extra na Adição de categorias

`do_action(insert_custom_dynatree_edit_category)`

single.php

permite adicionar um dynatree personalizado no modal de Edição de categorias

`do_action(javascript metas_category)`

single.JS

permite adicionar um script que consiga setar valores dos novos campos adicionados no modal de Edição

7.Metas das categorias

`apply_filters(modificate_returned_metas_categories, array())`

category_model.php

trabalha com os metadados de uma categoria para retornar os dados de forma customizável, é passado um array com o array padrão vindo do banco de dados já finalizado e o array com todos os metadados encontrados (array de arrays)

8.Eventos de categoria

do_action(after_event_edit_term)

event_term_edit_model.php

executa apos a execucao de um evento de Edição de categoria

9.Controlador - Propriedades

apply_filters(modificate_values_event_property_data_add, \$data)

property_controller.php

filtro que trabalha com os dados do formulário de Adição de propriedade de dados para os eventos

apply_filters(modificate_values_event_property_data_update, \$data)

property_controller.php

filtro que trabalha com os dados do formulário de Edição de propriedade de dados para os eventos

apply_filters(modificate_values_event_property_object_add, \$data)

property_controller.php

filtro que trabalha com os dados do formulário de Adição de propriedade de objetos para os eventos

10.Modelo - Propriedades

apply_filters(modificate_values_get_all_property, \$data)

property_model.php

filtro que trabalha com os dados ao editar uma propriedade

11.Eventos de Propriedade

do_action(after_event_add_property_data)

event_property_data_create

executa após um evento de Adição deste tipo de propriedade

do_action(after_event_update_property_data)

event_property_data_create

executa após um evento de atualização deste tipo de propriedade

do_action(after_event_delete_property_data)

event_property_data_create

executa após um evento de remoção deste tipo de propriedade

apply_filters(modificate_values_event_property_object_update, \$data)

property_controller.php

filtro que trabalha com os dados do formulário de Edição de propriedade de objeto para os eventos

12.Tela de metadados

Javascript - DATA

do_action(javascript_set_new_fields_edit_property_data)

list_js.php

permite adicionar um script que consiga setar valores na Edição de propriedade de dados

do_action(javascript_clear_forms)

list_js.php

permite adicionar um script que limpa o formulário de Adição/Edição de propriedades

DATA

do_action(form_help_property_data_type_[tipo])

list.php (properties)

permite a remoção dos tipos de metadados de dados

do_action(form_help_property_data_insert_types)

list.php (properties)

permite Adição de tipos para metadados de dados

do_action(form_modify_property_data)

list.php (properties)

permite a Adição de html no formulário de inserção propriedade de dados

do_action(form_default_value_property_data)

list.php (properties)

permite a remoção do campo de valor default

do_action(form_help_property_data)

list.php (properties)

permite a remoção do campo de ajuda

do_action(form_required_property_data)

list.php (properties)

permite a remoção do campo obrigatório

Javascript - Objeto

do_action(javascript_set_new_fields_edit_property_object)

list_js.php

permite adicionar um script que consiga setar valores na Edição de propriedade de objeto

do_action(javascript_onselect_relationship_dynatree_property_object)

list_js.php

permite adicionar um script que consiga realizar uma ação no select do dynatree de relacionamentos

Objeto

do_action(form_is_reverse_property_object)

list.php (properties)

permite a remoção do campo de propriedade reversa

do_action(form_required_property_object)

list.php (properties)

permite a remoção do campo obrigatório

13.Exportação RDF

`apply_filters(personal_export_category_rdf, $slug)`

`rdf_controller.php`

filtro que trabalha com a exportação de uma categoria

`apply_filters(modificate_export_complete_collection_rdf, $slug)`

`rdf_controller.php`

filtro que trabalha com a exportação de uma coleção completa

`apply_filters(modificate_export_collection_rdf, $slug)`

`rdf_controller.php`

filtro que trabalha com a exportação de uma coleção simples

14.Criação da coleção

`do_action(before_facets,$category_root->term_id,$collection_id)`

`functions.php`

permite a inserção de propriedades na coleção quando a citada é criada

`apply_filters(category_root_as_facet, true)`

`functions.php`

filtro que bloqueia a criação da categoria raiz por default

`apply_filters(show_checkbox_facet,')`

`functions.php`

permite a Adição do checkbox para as facetas

`do_action(before_facets,$facet)`

`menu_left.php` (Filters)

permite realizar qualquer ação antes da montagem dos filtros, é passado os filtros como parâmetro

17.Home do item

`do_action(home_item_source_div)`

`list_single_object.php`

permite a Adição de atributos html para a div que comporta a fonte do item

`do_action(home_item_type_div)`

`list_single_object.php`

permite a Adição de atributos html para a div que comporta o tipo do item

`do_action(home_item_license_div)`

`list_single_object.php`

permite a Adição de atributos html para a div que comporta a licença do item

do_action(home_item_tag_div)

list_single_object.php

permite a Adição de atributos html para a div que comporta a tag do item

do_action(home_item_attachments_div)

list_single_object.php

permite a Adição de atributos html para a div que comporta os anexos do item

do_action(home_item_insert_container)

list_single_object.php

permite a Adição de atributos html para a div que comporta os anexos do item

18.inserção/Edição de itens: Propriedades

do_action(modificate_label_insert_item_properties)

show_insert_object_properties.php

gancho para alterar o label das propriedades na inserção de itens

do_action(modificate_label_edit_item_properties)

list_properties.php

gancho para alterar o label das propriedades na Edição de itens

do_action(modificate_insert_item_properties_data)

show_insert_object_properties.php

gancho para tipos de metadados de dados diferentes - adição de itens

do_action(modificate_edit_item_properties_data)
list_properties.php (Edit)

gancho para tipos de metadados de dados diferentes - edição de itens

do_action(modificate_insert_item_properties_object)
show_insert_object_properties.php

gancho para tipos de metadados de objetos diferentes - adição de itens

do_action(modificate_edit_item_properties_object)
list_properties.php

gancho para tipos de metadados de objetos diferentes - edição de itens

19.Home do item: Propriedades

do_action(modificate_single_item_properties_data)
show_insert_object_properties.php

gancho para tipos de metadados de dados diferentes - página do item

do_action(modificate_single_item_properties_object)
show_insert_object_properties.php

gancho para tipos de metadados de object diferentes - página do item

