

Drupalcamp

Spain 2012

Madrid, 20-21 octubre

El mayor evento
Drupal de España



TITLE: Field Types API: Field,
widgets y formatters

PRESENTERS: Miguel Ordóñez

Quien Soy?

- ⌘ Nombre: Miguel Ordóñez (mordonez)
- ⌘ Twitter: <http://twitter.com/miguelordonez>
- ⌘ Co-fundador de Ymbra

Agenda

- ⌄ Algunos conceptos
 - ⌄ Que es un Field
 - ⌄ Field API
- ⌄ Como crear tu propio field
 - ⌄ Schema
 - ⌄ Widget
 - ⌄ Formatter
 - ⌄ Validation
- ⌄ Algunas cosas a tener en cuenta
- ⌄ Drupal 8
- ⌄ Referencias

Algunos conceptos

Algunos conceptos: Que es un Field

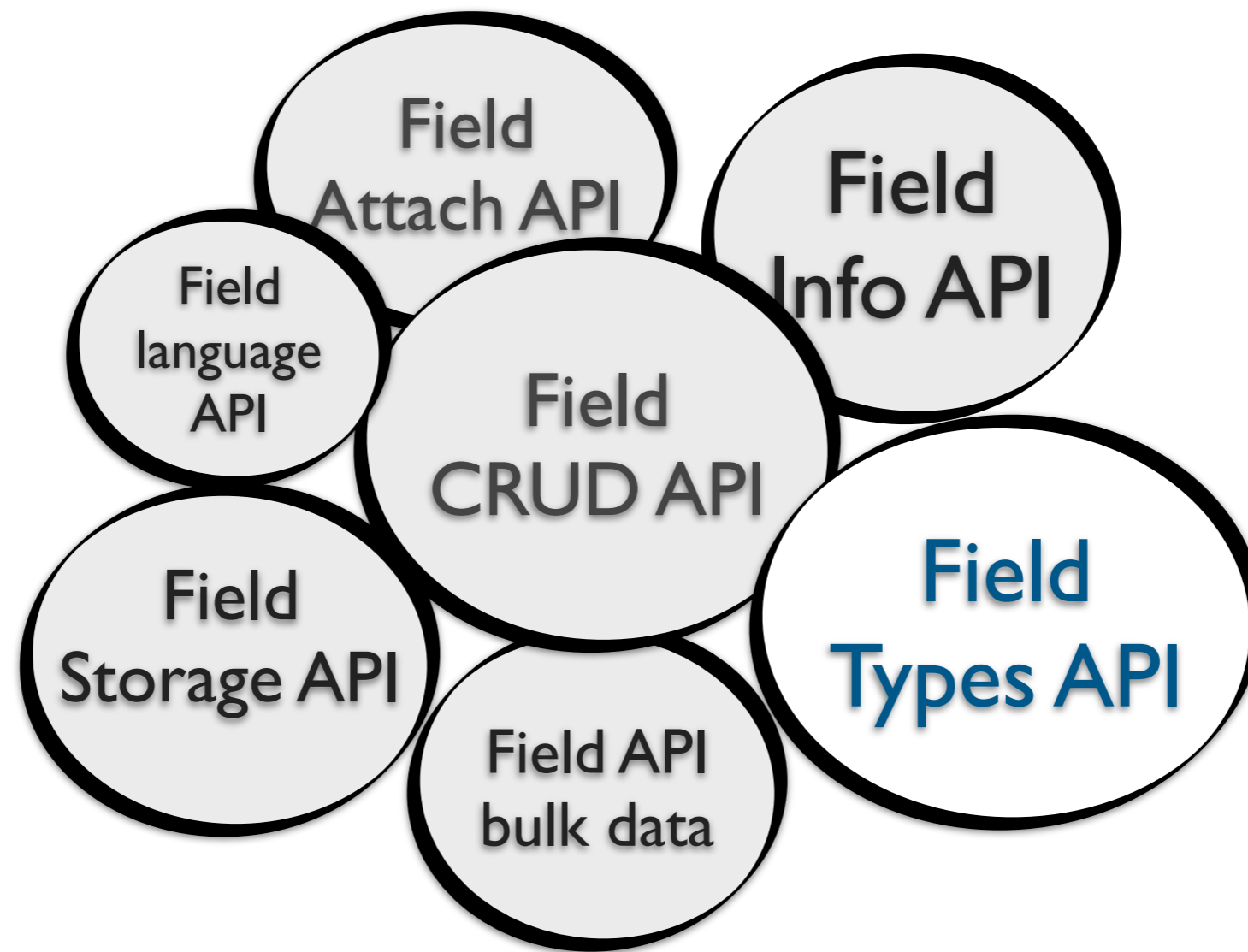
Es una pieza de funcionalidad que se adjunta a una entidad (nodo, taxonomia, usuario, etc...)



The image shows a screenshot of a Drupal form interface. At the top, there is a section titled "DATE" with an "All Day" checkbox. Below this, there are two input fields: "Date" containing "09/20/2011" and "Time" containing "18:15". A calendar widget is open, showing the month of September 2011. The calendar grid has days of the week (MO, TU, WE, TH, FR, SA, SU) as columns and dates as rows. The date "20" is highlighted. Below the calendar, there are labels for "Title" and "Blog" with corresponding input fields.

Algunos conceptos: Field API

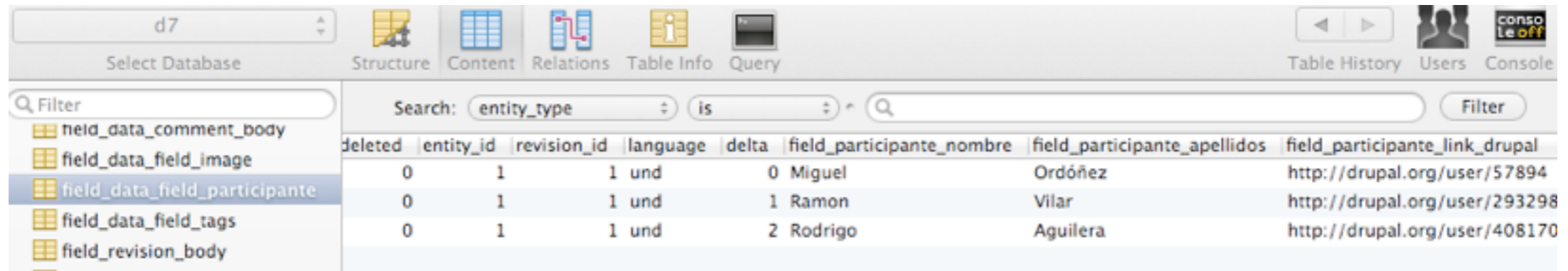
“The Field API allows custom data fields to be attached to Drupal entities and takes care of storing, loading, editing, and rendering field data” (<http://api.drupal.org>)



Algunos conceptos: Field Type, Widget, Formatter

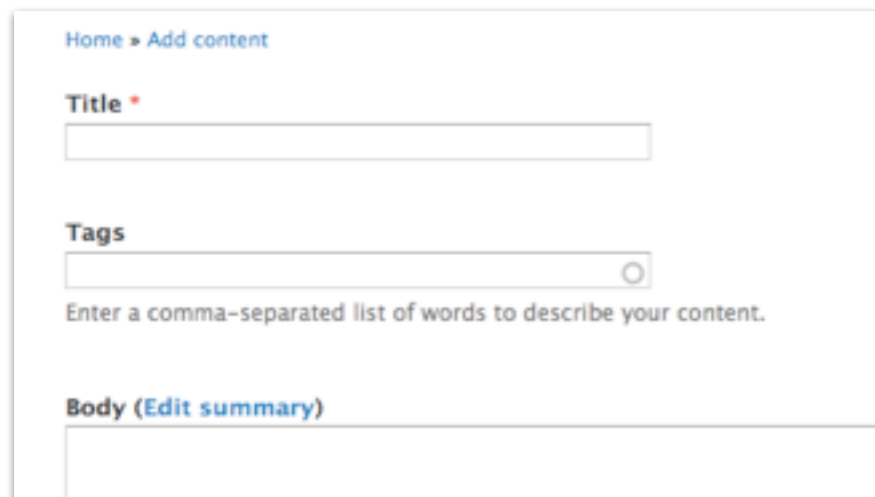
⇒ Un campo puede tener 3 piezas importantes

- Field type.



deleted	entity_id	revision_id	language	delta	field_participante_nombre	field_participante_apellidos	field_participante_link_drupal
0	1	1	und	0	Miguel	Ordóñez	http://drupal.org/user/57894
0	1	1	und	1	Ramon	Vilar	http://drupal.org/user/293298
0	1	1	und	2	Rodrigo	Aguilera	http://drupal.org/user/408170

- Widget.



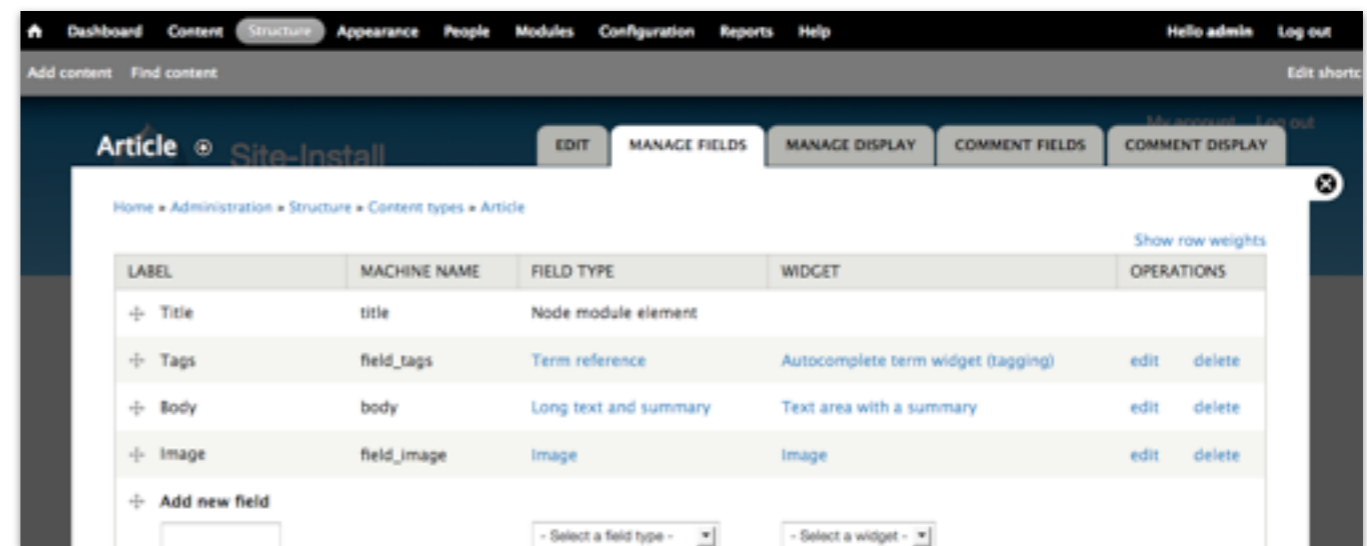
Home » Add content

Title *

Tags

Enter a comma-separated list of words to describe your content.

Body (Edit summary)



LABEL	MACHINE NAME	FIELD TYPE	WIDGET	OPERATIONS
↕ Title	title	Node module element		
↕ Tags	field_tags	Term reference	Autocomplete term widget (tagging)	edit delete
↕ Body	body	Long text and summary	Text area with a summary	edit delete
↕ Image	field_image	Image	Image	edit delete
↕ Add new field				

Algunos conceptos: Field Type, Widget, Formatter

🔗 y el Formatter.

Tags:
[Drupal](#) [Article](#) [Drupal Planet](#)

FIELD	LABEL	FORMAT	
✚ Image	<Hidden>	Image	Image style: large
✚ Body	<Hidden>	Default	
✚ Tags	Above	Link	
Hidden			
No field is hidden.			

Ejemplo, Como crear un campo personalizado

Ejemplo, Como crear un campo personalizado

PARTICIPANTE:

Nombre

Apellidos

Perfil en Drupal.org

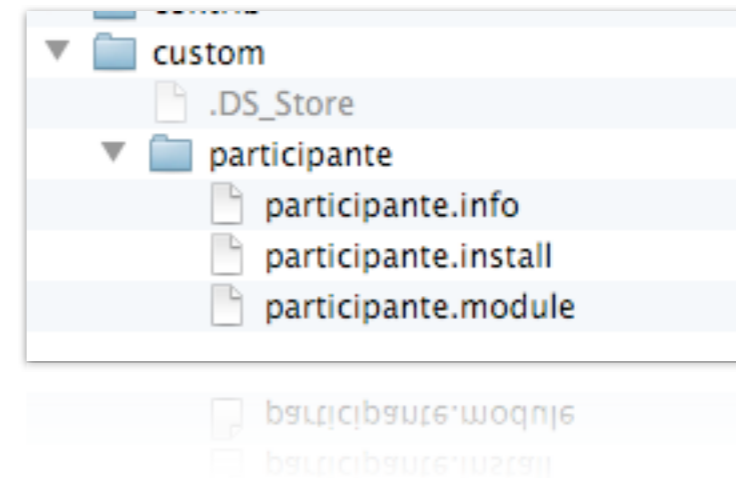
Example: <http://drupal.org/user/57894>

Add another item

- ⦿ Vamos a crear un campo absurdo como ejemplo, lo utilizaremos para añadir participantes de la drupalcamp.
- ⦿ El campo tendrá los campos: nombre, apellidos y el enlace al perfil de drupal.org
- ⦿ Podremos decidir si queremos enlazar el participante a su perfil de drupal.

Ejemplo, Como crear un campo personalizado

- ⇒ Creamos el módulo “participante”
- ⇒ Creamos los ficheros .info, .install i .module



Ejemplo, Como crear un campo personalizado: .info

🔗 participante.info

```
1 name = Participante Field Demo
2 description = Un ejemplo de un tipo de campo personalizado para la drupalCamp 2012
3 package = dc
4 core = 7.x
5 dependencies[] = field_ui
6 files[] = participante.module
7
```

Ejemplo, Como crear un campo personalizado: Field Type

⌚ participante.module (**hook_field_info**)

```
7  /**
8   * Implements hook_field_info().
9   */
10 function participante_field_info() {
11     return array(
12         'participante' => array(
13             'label' => t('Participante'),
14             'description' => t('Informacion de un participante'),
15             'default_widget' => 'participante_participante_form',
16             'default_formatter' => 'participante_default',
17         ),
18     );
19 }
```

* los literales mejor en **inglés**. Pero para el ejemplo se pusieron en español :-P

Ejemplo, Como crear un campo personalizado: Field Type

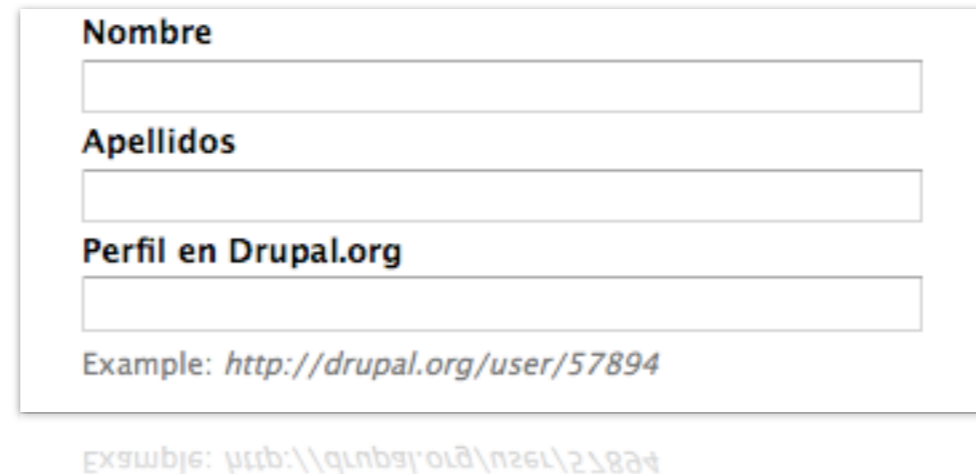
↻ participante.install (hook_field_schema)

```
23  */
24  function participante_field_schema($field) {
25    if ($field['type'] == 'participante') {
26      $columns = array(
27        'nombre' => array(
28          'type' => 'varchar',
29          'length' => 255,
30          'not null' => FALSE
31        ),
32        'apellidos' => array(
33          'type' => 'varchar',
34          'length' => 255,
35          'not null' => FALSE
36        ),
37        'link_drupal' => array(
38          'type' => 'varchar',
39          'length' => 255,
40          'not null' => FALSE
41      ),
42    );
43
44    $indexes = array(
45      'nombre' => array('nombre'),
46      'apellidos' => array('apellidos'),
47    );
48
49    return array(
50      'columns' => $columns,
51      'indexes' => $indexes,
52    );
53  }
54 }
```

Schema API:
drupal.org/node/146843

Ejemplo, Como crear un campo personalizado: **Widget**

- ⌚ Crear el **Widget** d'entrada.



Nombre

Apellidos

Perfil en Drupal.org

Example: <http://drupal.org/user/57894>

- ⌚ Definimos la información básica del widget por medio **hook_field_widget_info**
- ⌚ Utilizamos **hook_field_widget_form** para definir el formulario d'entrada del campo.
- ⌚ Utilizamos **hook_field_is_empty** para validar que nuestro campo no está vacío.

Ejemplo, Como crear un campo personalizado: Widget

hook_field_widget_info

```
33 /**
34  * Implements hook_field_widget_info().
35  */
36 function participante_field_widget_info() {
37     return array(
38         'participante_participante_form' => array(
39             'label' => t('Formulario del participante'),
40             'field types' => array('participante'),
41         ),
42     );
43 }
44
```


Ejemplo, Como crear un campo personalizado: Widget

```
44
45 /**
46  * Implements hook_field_widget_form().
47  */
48 function participante_field_widget_form(&$form, &$form_state, $field, $instance,
49   $langcode, $items, $delta, $element) {
50
51   $base = $element;
52   switch ($instance['widget']['type']) {
53     case 'participante_participante_form':
54       $widget = $instance['widget'];
55       $settings = $widget['settings'];
56
57       $element['nombre'] = array(
58         '#type' => 'textfield',
59         '#title' => t('Nombre'),
60         '#default_value' => isset($items[$delta]['nombre']) ?
61           $items[$delta]['nombre'] : NULL,
62       );
63       $element['apellidos'] = array(
64         '#type' => 'textfield',
65         '#title' => t('Apellidos'),
66         '#default_value' => isset($items[$delta]['apellidos'])
67           ? $items[$delta]['apellidos'] : NULL,
68       );
69       $element['link_drupal'] = array(
70         '#type' => 'textfield',
71         '#title' => t('Perfil en Drupal.org'),
72         '#description' => t('Example: <em>http://drupal.org/user/57894</em>'),
73         '#default_value' => isset($items[$delta]['link_drupal'])
74           ? $items[$delta]['link_drupal'] : NULL,
75         '#element_validate' => array('_participante_field_link_validate'),
76       );
77       break;
78   }
79
80   return $element;
81
82
```

Ejemplo, Como crear un campo personalizado: Widget

- Para añadir una validación, utilizaremos `#element_validate`

```
69     $element['link_drupal'] = array(  
70         '#type' => 'textfield',  
71         '#title' => t('Perfil en Drupal.org'),  
72         '#description' => t('Example: <em>http://drupal.org/user/57894</em>'),  
73         '#default_value' => isset($items[$delta]['link_drupal'])  
74             ? $items[$delta]['link_drupal'] : NULL,  
75         '#element_validate' => array('_participante_field_link_validate'),  
76     );  
77     break;  
78 }  
79  
80 return $element;  
81 }  
82  
83 /**  
84  * Validacin callback para un enlace al perfil de drupal.org.  
85  */  
86 function _participante_field_link_validate($element, &$form_state, $form) {  
87     $value = $element['#value'];  
88  
89     if (!empty($value) && !valid_url($value, TRUE)) {  
90         form_error($element, t('Invalid URL.'));  
91     }  
92 }  
93  
94 }  
95 }  
96 }  
97 }  
98 }  
99 }  
100 }  
101 }  
102 }  
103 }  
104 }  
105 }  
106 }  
107 }  
108 }  
109 }  
110 }  
111 }  
112 }  
113 }  
114 }  
115 }  
116 }  
117 }  
118 }  
119 }  
120 }  
121 }  
122 }  
123 }  
124 }  
125 }  
126 }  
127 }  
128 }  
129 }  
130 }  
131 }  
132 }  
133 }  
134 }  
135 }  
136 }  
137 }  
138 }  
139 }  
140 }  
141 }  
142 }  
143 }  
144 }  
145 }  
146 }  
147 }  
148 }  
149 }  
150 }  
151 }  
152 }  
153 }  
154 }  
155 }  
156 }  
157 }  
158 }  
159 }  
160 }  
161 }  
162 }  
163 }  
164 }  
165 }  
166 }  
167 }  
168 }  
169 }  
170 }  
171 }  
172 }  
173 }  
174 }  
175 }  
176 }  
177 }  
178 }  
179 }  
180 }  
181 }  
182 }  
183 }  
184 }  
185 }  
186 }  
187 }  
188 }  
189 }  
190 }  
191 }  
192 }  
193 }  
194 }  
195 }  
196 }  
197 }  
198 }  
199 }  
200 }
```

Ejemplo, Como crear un campo personalizado: Widget

hook_field_is_empty

```
21  /**
22   * Implements hook_field_is_empty().
23   */
24  function participante_field_is_empty($item, $field) {
25      if ($field['type'] == 'participante') {
26          if (empty($item['nombre']) && empty($item['apellidos'])) {
27              return TRUE;
28          }
29      }
30      return FALSE;
31  }
32
33
34  }
35  return FALSE;
36  }
```

Ejemplo, Como crear un campo personalizado: Formatter

- ⌚ Crearemos el **formatter** para mostrar los valores del campo.



- ⌚ Utilizaremos **hook_field_formatter_info** para definir el formatter.
- ⌚ Utilizaremos **hook_field_formatter_view** para definir la lógica de salida.

Ejemplo, Como crear un campo personalizado: Formatter

hook_field_formatter_info

```
94  /**
95   * Implements hook_field_formatter_info().
96   */
97  function participante_field_formatter_info() {
98      return array(
99          'participante_default' => array(
100             'label' => t('Default'),
101             'field types' => array('participante'),
102             'settings' => array(
103                 'link' => TRUE,
104             ),
105         ),
106     );
107 }
```

Ejemplo, Como crear un campo personalizado: Formatter

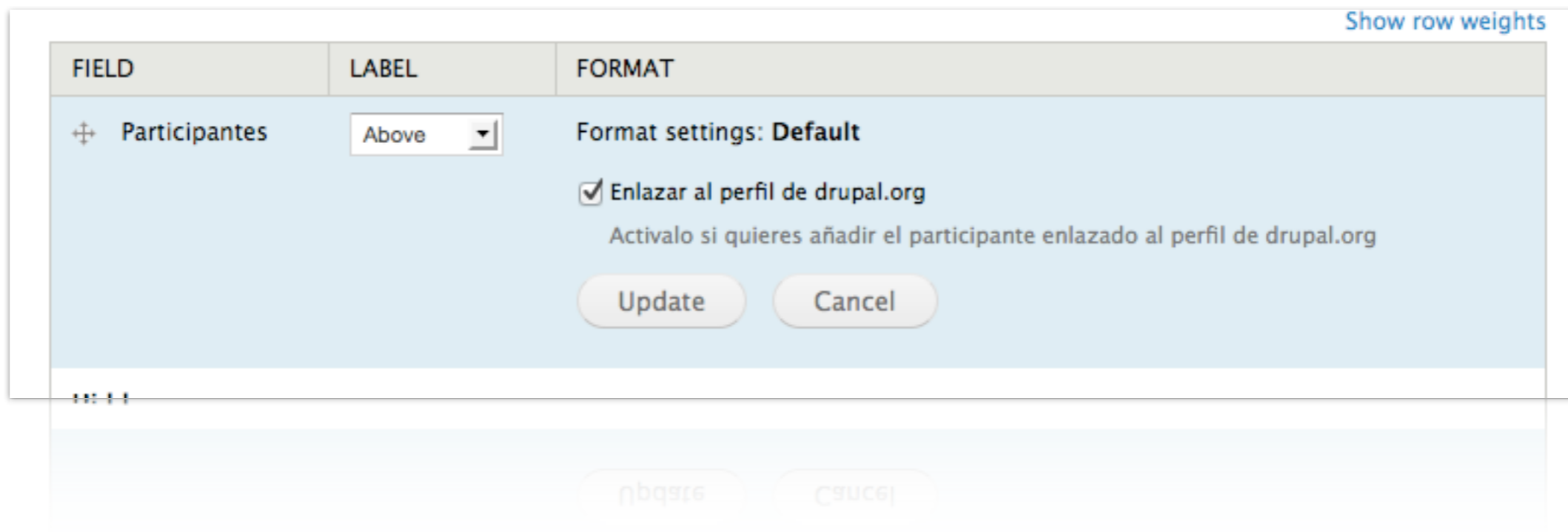
hook_field_formatter_view

```
147 /**
148  * Implements hook_field_formatter_view().
149  */
150 function participante_field_formatter_view($entity_type, $entity, $field,
151   $instance, $langcode, $items, $display) {
152   $element = array();
153   $settings = $display['settings'];
154
155   switch ($display['type']) {
156     case 'participante_default':
157       foreach ($items as $delta => $item) {
158         $value = $item['nombre'] . ' ' . $item['apellidos'];
159
160         $element[$delta] = array(
161           '#type' => 'html_tag',
162           '#tag' => 'p',
163           '#value' => $settings['link'] ? l($value, $item['link_drupal']) : $value,
164         );
165       }
166       break;
167     }
168
169   return $element;
170 }
```

Ejemplo, Como crear un campo personalizado: Formatter

Utilizaremos

hook_field_formatter_settings_summary y **hook_field_formatter_settings_form** para añadir elementos configurables del formatter.



The screenshot shows the 'Format settings' for a field named 'Participantes'. The interface is divided into three columns: 'FIELD', 'LABEL', and 'FORMAT'. The 'FIELD' column contains a plus icon and the text 'Participantes'. The 'LABEL' column contains a dropdown menu with 'Above' selected. The 'FORMAT' column contains the text 'Format settings: Default', a checked checkbox for 'Enlazar al perfil de drupal.org', and a subtext 'Activalo si quieres añadir el participante enlazado al perfil de drupal.org'. At the bottom of the settings area are two buttons: 'Update' and 'Cancel'. A 'Show row weights' link is visible in the top right corner of the table area.

FIELD	LABEL	FORMAT
⊕ Participantes	Above	Format settings: Default <input checked="" type="checkbox"/> Enlazar al perfil de drupal.org Activalo si quieres añadir el participante enlazado al perfil de drupal.org Update Cancel

Ejemplo, Como crear un campo personalizado: Formatter

hook_field_formatter_settings_form

```
125 /**
126  * Implements hook_field_formatter_settings_form().
127  */
128 function participante_field_formatter_settings_form($field, $instance, $view_mode,
129   $form, &$form_state) {
130   $display = $instance['display'][$view_mode];
131   $settings = $display['settings'];
132
133   if ($display['type'] == 'participante_default') {
134
135     $element['link'] = array(
136       '#type' => 'checkbox',
137       '#title' => t('Enlazar al perfil de drupal.org'),
138       '#default_value' => $settings['link'],
139       '#required' => FALSE,
140       '#description' => t('Activalo si quieres añadir el participante
141         enlazado al perfil de drupal.org'),
142     );
143   }
144
145   return $element;
146 }
```


Ejemplo, Como crear un campo personalizado: Formatter

hook_field_formatter_settings_summary

```
109  /**
110   * Implements hook_field_formatter_settings_summary().
111   */
112  function participante_field_formatter_settings_summary($field, $instance,
113    $view_mode) {
114
115    $display = $instance['display'][$view_mode];
116    $settings = $display['settings'];
117    $summary = '';
118
119    if ($display['type'] == 'participante_default') {
120      $summary = $settings['link'] ?
121        t('Enlazado al perfil de drupal.org') : t('Sin enlazar');
122    }
123
124    return $summary;
125  }
```

Ejemplo, Como crear un campo personalizado: Formatter

- Algunos ejemplos de formatters: <http://customformatters.com>

The screenshot shows the 'CUSTOM FORMATTERS' website with the tagline 'FORMAT ALL THE FIELDS'. It features a navigation menu with 'FORMATTERS' and 'BLOG' links. The main content area displays an example titled 'EXAMPLE: IMAGE (HTML + TOKEN)' with the machine name 'EXAMPLE_TOKEN_IMAGE'. Below the title, there are tabs for 'TOKEN' and 'IMAGE'. A descriptive paragraph states: 'This is a HTML + Tokens example formatter provided by the Custom Formatters module that displays a thumbnail image linked to a large image.' To the right of the main content, there are several buttons: 'DOWNLOAD 7.x-2.2' and 'STEAL THIS SITE @ GITHU'. A large, vertical watermark 'FORMATTERS' is overlaid on the right side of the page. At the bottom of the screenshot, a code block shows the PHP implementation of the formatter:

```
1. /**
2.  * Implements hook_field_formatter_info().
3.  */
4. function MYMODULE_field_formatter_info() {
5.   return array(
6.     'MYMODULE_example_token_image' => array(
7.       'label' => t('Example: Image (HTML + Token)'),
8.       'field types' => array('image'),
9.     ),
10.  );
11. }
12.
13. /**
14.  * Implements hook_field_formatter_info().
15.  */
16. }
```

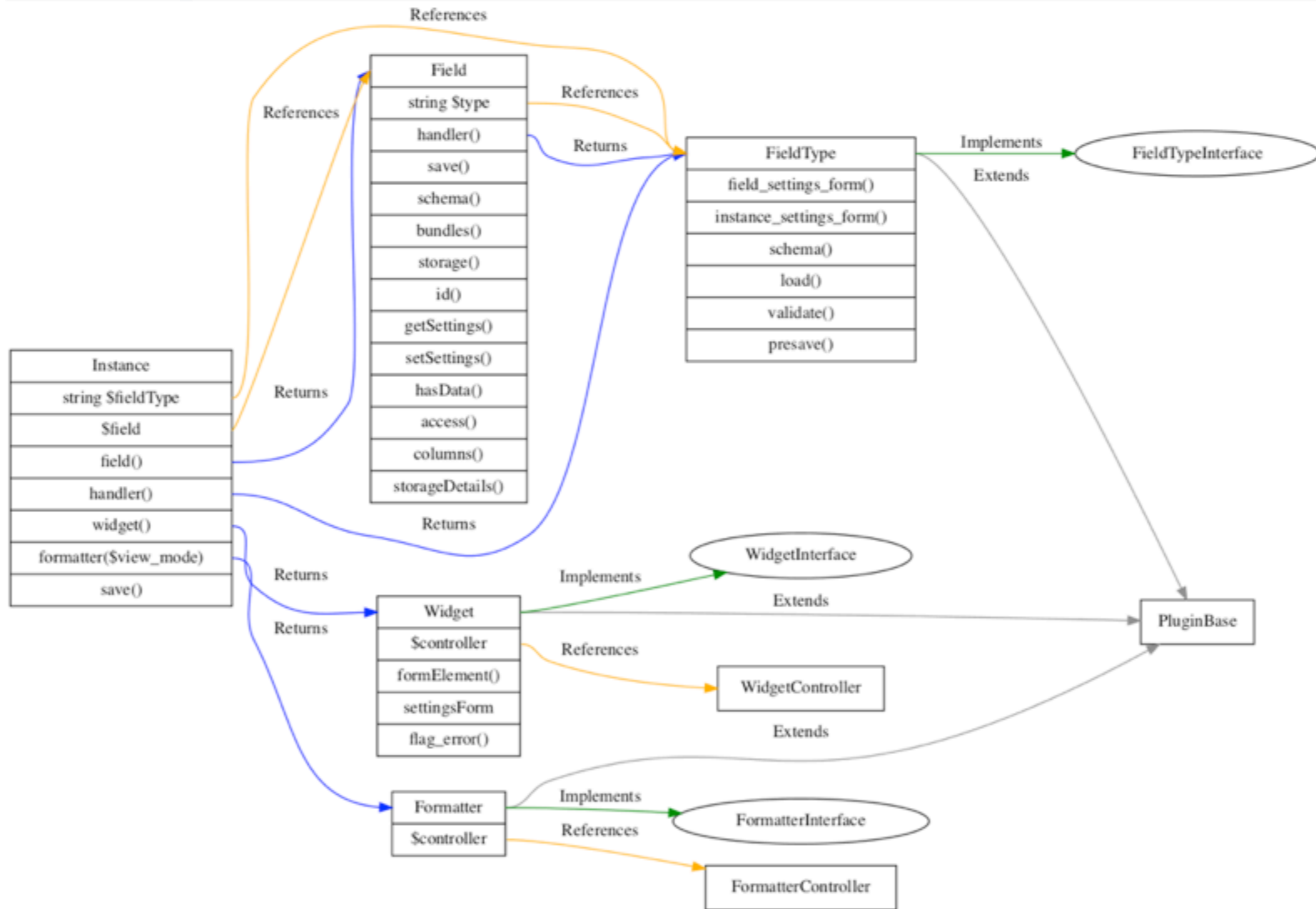
Algunas cosas a tener en cuenta

- ↻ Un Field puede tener más de un formatter o widget a la vez.
- ↻ Se pueden crear campos personalizados sin crear widgets y viceversa.

Drupal 8?

- ↻ CMI
- ↻ Widget y formatter como plugins en drupal 8
- ↻ Field API vs OO

Drupal 8?



* Overview of the pluginification of Field API in Drupal 8: <http://drupalsun.com/node/32450>

Referencias

- ⌘ <http://api.drupal.org>
- ⌘ Field Types API: http://api.drupal.org/api/drupal/modules--field--field.api.php/group/field_types/7
- ⌘ Examples for Developers: <http://drupal.org/project/examples>
- ⌘ <http://customformatters.com>
- ⌘ Overview of the pluginification of Field API in Drupal 8, <http://drupalsun.com/node/32450>
- ⌘ Poutine Maker: An Introduction to the Field API in Drupal 7 (Part I) <http://evolvingweb.ca/story/poutine-maker-introduction-field-api-drupal-7-part-1>
- ⌘ [Field API - Field UI - Drupal 8](#)
- ⌘ Form API Reference. http://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7

MUCHAS GRACIAS