Drupal 7 visto a través de Drupal Commerce

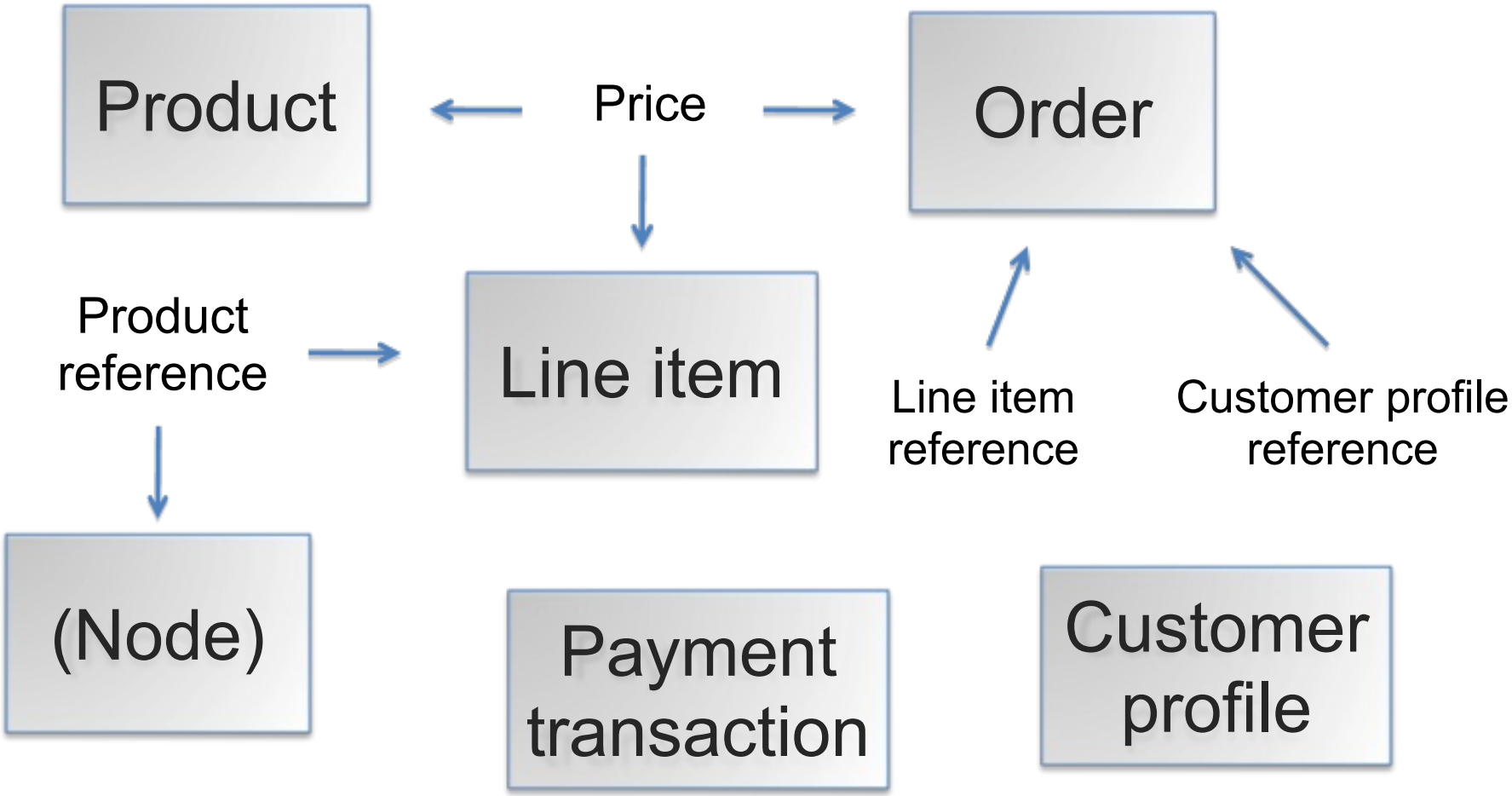
commerce guys

¿Qué es Drupal Commerce?

Drupal Commerce es un nuevo framework de comercio electrónico desarrollado de forma nativa para **Drupal 7** con el objetivo de hacer de Drupal un referente en el mundo del ecommerce open source.

commerce guys

Componentes de Commerce



commerce guys

Separación de Producto y Display

Los productos están separados en una parte de **display** (nodos) y una nueva **entidad** “product” que se relacionan a través de un campo “product reference”.

Esto supone una gran ventaja de cara a la **arquitectura y flexibilidad** pero la usabilidad por defecto se ve afectada.

commerce guys

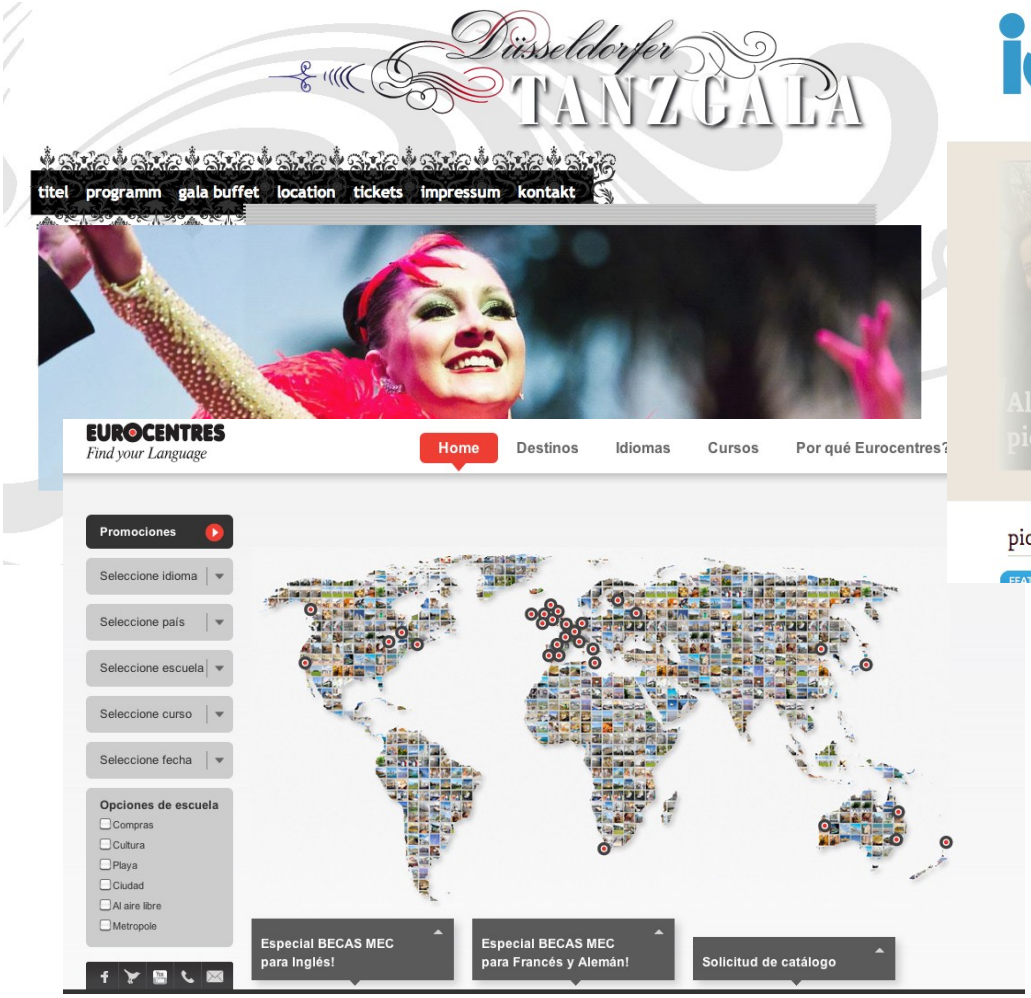
Extensiones disponibles

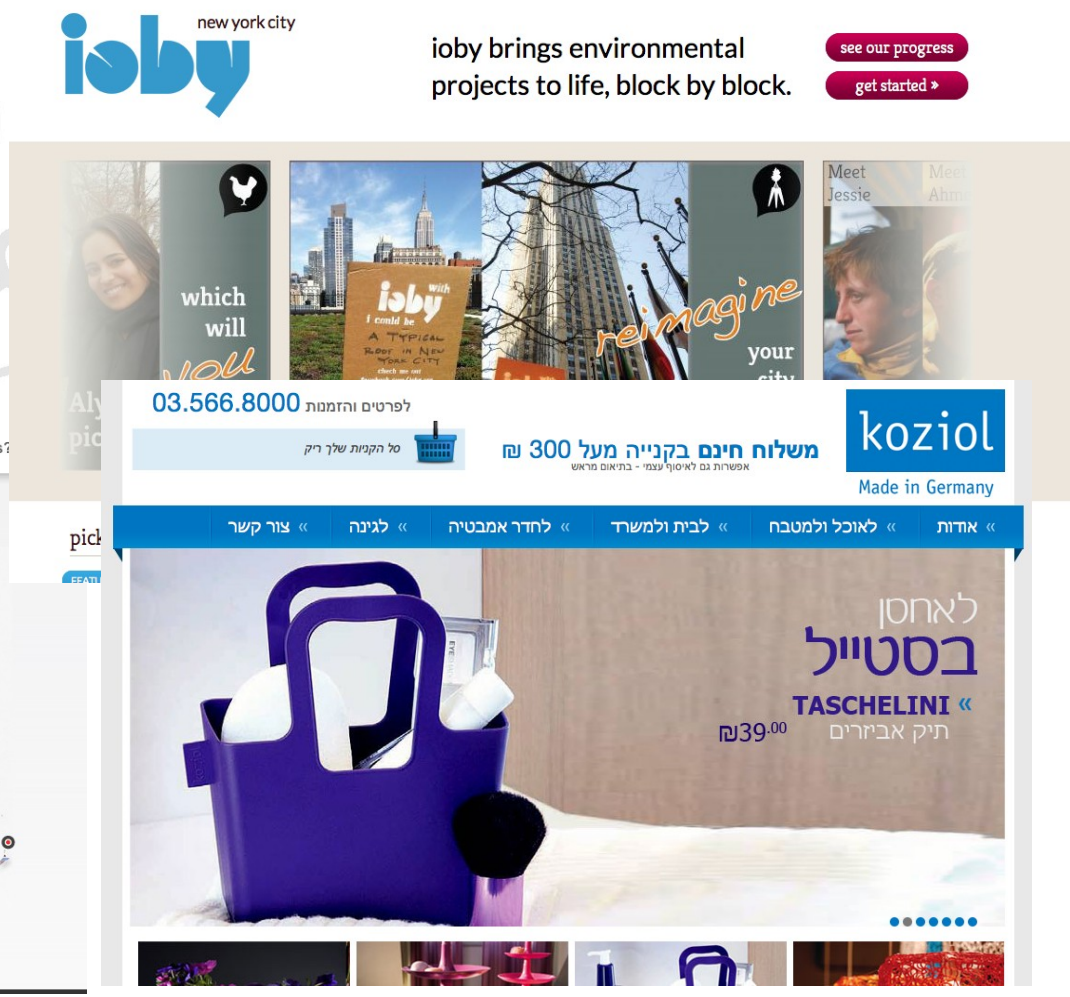
- **Migración**: Commerce Feeds, Commerce Migrate
- **Usabilidad**: Commerce Product Display Manager.
- **Utilidades**: Commerce File, Commerce Coupon, Commerce Google Analytics, Commerce Stock, Commerce Shipping
- **Pago**: Commerce Paypal, Commerce Sermepa

Más: http://www.drupalcommerce.org/contrib

commerce guys

Ejemplos de tiendas con Commerce





Más: http://www.drupalcommerce.org/showcase

commerce guys

¿Demo?

http://bit.ly/dc-demo-cg

commerce guys

Concepto de Entity

Las entidades añaden una **nueva capa de abstracción** basada en objetos sobre los datos que permite que todo el contenido comparta API y workflow

Nodos, usuarios, términos de taxonomía, vocabularios, comentarios, ficheros... son ahora entidades

"~~Everything is a node~~" vs "**Nodes are entities too**"

commerce guys

Ejemplo #1: Entity Controllers

```
class CommerceProductEntityController extends
DrupalCommerceEntityController {
(..)
  public function create(array $values = array()) {
    return (object) ($values + array(
      'product_id' => '',
      'is_new' => TRUE,
      'sku' => '',
      'title' => '',
      'uid' => '',
      'status' => 1,
      'created' => '',
      'changed' => '',
    ));
  }
(..)
}
```

commerce guys

Ejemplo #2: EntityFieldQuery

```
function
commerce_product_reference_commerce_product_can_delete($product) {
  // Use EntityFieldQuery to look for line items referencing this
  // product and do not allow the delete to occur if one exists.
  $query = new EntityFieldQuery();

  $query
    ->entityCondition('entity_type', 'commerce_line_item', '=')
    ->entityCondition('bundle', 'product', '=')
    ->fieldCondition('product', 'product_id',
        $product->product_id, '=')
    ->count();

  return $query->execute() > 0 ? FALSE : TRUE;
}
```

commerce guys

Entities como base de Commerce

- **Entidades** en Drupal Commerce: Orders, Products, Line items, Transactions, Profiles.
- La capacidad de añadir campos a entidades (propiedad *fieldable*) permite extenderlas con diferentes datos asociados.
- Es posible definir otras entidades para casos concretos, por ejemplo Commerce Delivery o Commerce Shipping

commerce guys

Entity API

- El módulo Entity API se crea para facilitar el acceso a las entidades y para rellenar los huecos que le faltan al core de Drupal.
- Añade elementos muy interesantes como las propiedades, exportables, interfaz de administración o los *metadata wrappers*
- También proporciona un controlador de CRUD estándar para la mayoría de entidades.

commerce guys

Ejemplo: Entity Metadata Wrapper

```
function commerce_line_items_quantity($line_items, $type = '') {
  // Sum up the quantity of all matching line items.
  $quantity = 0;

  foreach ($line_items as $line_item) {
    if (!is_a($line_item, 'EntityMetadataWrapper')) {
      $line_item = entity_metadata_wrapper('commerce_line_item',
          $line_item);
    }

    if (empty($type) || $line_item->type->value() == $type) {
      $quantity += $line_item->quantity->value();
    }
  }

  return $quantity;
}
```

commerce guys

Field API

- Parte del módulo de CCK se ha refactorizado y es ahora el Field API de Drupal 7.
- Los nuevos campos generados por Field API se pueden adjuntar a cualquier entidad que tenga la propiedad *fieldable*.
- Se ha facilitado en gran medida la forma de crear campos personalizados y se han añadido y estandarizado gran cantidad de nuevas opciones.

commerce guys

Extendiendo el Field API

Field API incorpora gran cantidad de hooks, ejemplos curiosos:

- hook_field_attach_* - controlan los formularios y acciones cuando un campo se adjunta a una entidad (y operaciones CRUD).
- hook_field_storage_* - controlan la forma en la que se almacena el campo (NoSQL?)
- hook_field_extra_fields* - Permiten exponer “pseudo campos” en las entidades.

commerce guys

Interfaz de usuario basada en Views

Drupal Commerce carece de pantallas de administración y gestión propias, todos los listados se basan en el módulo **Views 3** y son totalmente configurables... y opcionales.

commerce guys

Novedades de Views 3

"Todo" se convierte en un plugin de **CTools**.

- Query Backend.
- Traducción.
- Areas: Header, Footer, Empty text.

Interfaz reconstruida totalmente.

- Filtros Contextuales.
- Semantic Views.
- Exposed Sorts.
- Más usable.

commerce guys

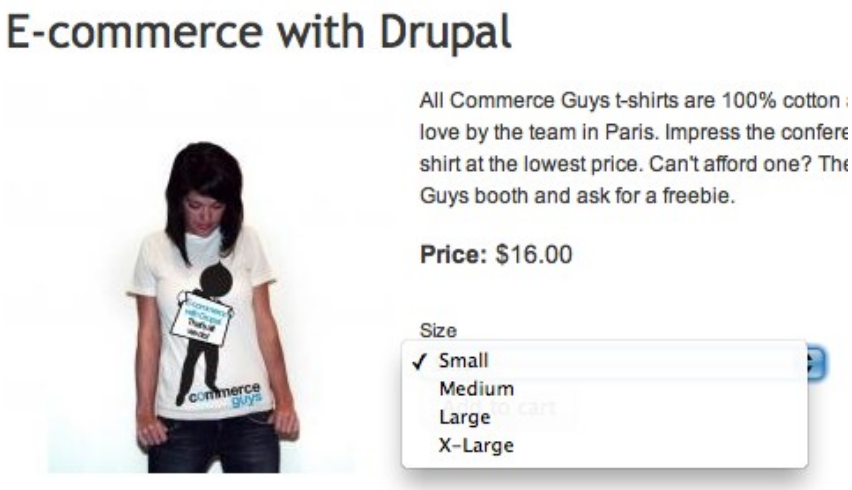
Product References, Prices...

Drupal Commerce hace un uso extensivo del nuevo Field API.

- Los productos se relacionan con los “Product Displays” a través de un campo “Product Reference”
- Los pedidos se relacionan con las líneas de pedido a través de un campo “Line item Reference”.
- El formulario de añadir al carrito es un Display Formatter de un campo de tipo precio.

commerce guys

“Add to cart” como Formatter



http://demo.commerceguys.com/dc/catalog/wearables/e-commerce-drupal

commerce guys

Perfiles de usuario extensibles

- Los perfiles de usuario o “Customer Profiles” son entidades a las que se les puede añadir campos.
- Se asocian a cada usuario que hace una compra.
- La base de los perfiles de usuario es el nuevo tipo de campo dirección, AddressField.
- Además de la dirección, el perfil de usuario puede contener cualquier información, NIF, Teléfono, que se puede añadir desde interfaz.

commerce guys

Addressfield

"*Este módulo solo hace una cosa, pero la hace bien: Almacena direcciones postales internacionales*"

- Provee soporte de direcciones para los customer profiles de Commerce.
- Módulo para Drupal 7 que almacena direcciones postales en el formato estándar xNAL (Usado por google maps para geocoding).

commerce guys

Workflow con Rules

Si el interfaz de Drupal Commerce está basado en Views, todas las opciones de flujo de trabajo se basan en **Rules**.
Modificaciones de precios como **descuentos** o **impuestos** están basados en Rules y por ello son totalmente configurables, lo mismo pasa con el checkout, el proceso de pago y customer profiles

commerce guys

Rules para Drupal 7

Rules 2.x para Drupal 7 es una reescritura de Rules para Drupal 6, tanto en interfaz como en proceso, ya que se basa totalmente en **Entity API**.

- Loops en Rules.
- Rules scheduler.
- “Entity agnostic”

commerce guys

Sistema de pagos transaccional

Un ejemplo de Rules y Entities en Drupal Commerce es el sistema de pagos por **transacciones**.
Los pagos quedan registrados en una entidades llamadas **Payment Transactions** que registran el balance de compra a través de Rules, es decir, el método de pago es una Rule.
Se pueden lanzar acciones de Rules adicionales antes/después de los pagos.

commerce guys

Novedades de Form API

- #states
- #ajax
- #attached
- hook_form_alter() desde plantilla
- #title_display
- #limit_validation_errors
- Vertical Tabs
- machine_name
- tableselect
- managed_file
- Elementos HTML5 a través del módulo Elements

commerce guys

Commerce “sin” javascript propio

Drupal Commerce apenas tiene unas pocas líneas de js propio, se apoya en gran medida en #states y #ajax

```
$form['attributes'][$field_name] = array(
    '#type' => 'select',
    '#title' => check_plain($data['instance']['label']),
    '#options' => array_intersect_key($data['options'],
          drupal_map_assoc($used_options[$field_name])),
    '#default_value' => $default_product_wrapper->{$field_name}
          ->value(),
    '#weight' => $data['instance']['widget']['weight'],
    '#ajax' => array(
       'callback' =>
          'commerce_cart_add_to_cart_form_attributes_refresh',
              ),
);
```

commerce guys

Capa de abstracción de BBDD

El proyecto **DBTNG** ha reformado la capa de abstracción de Drupal para acercarla a un modelo OOP.

```
$result = db_query("SELECT n.nid, u.name
                    FROM {node} n
                    WHERE n.type = '%s'
                    AND n.status = %d",
          array('page', 1));
```

Drupal 6

```
$product_count = db_select('commerce_product', 'cp')
  ->fields('cp', array('product_id'))
  ->countQuery()
  ->execute()
  ->fetchField();
```

Drupal 7

commerce guys

Tests unitarios y de integración

Drupal 7 incorpora el módulo Simpletest en su núcleo y tiene **unit testing** e **integration testing** en todos los componentes por defecto.
Drupal Commerce incorpora **integration testing** para todos sus componentes y empieza a integrar unit testing.
Tener los elementos verificados proporciona **seguridad** y **calidad** en los componentes.

commerce guys

Recursos Drupal 7

- Artículo sobre Entities: parte 1, 2 y 3.
- Ejemplo de EntityFieldQuery.
- Presentación de Fago sobre Entity API.
- Documentación oficial de Field API.
- Roadmap de Views 3.
- Videocasts sobre Rules de NodeOne.
- Forms API Reference.

commerce guys

Recursos Drupal Commerce

http://www.drupalcommerce.org

http://drupal.org/project/commerce

http://drupal.org/project/commerce_kickstart

http://www.commerceguys.com/resources

http://twitter.com/drupalcommerce

commerce guys

¿Preguntas?

- @pcambra
- cambrico.net
- Perfil en Drupal.org

commerce guys

¡Muchas gracias!

commerce guys