UD7 – Internacionalización y localización

2º CFGS
Desarrollo de Aplicaciones Web

2022-23

1.- Internacionalización y localización

Se conoce como **internacionalización** al proceso de diseñar software adaptado a diferentes regiones.

La **localización** es el proceso de configurar para una región determinada un software internacionalizado

2.- Localización

La internacionalización y la localización cubre, entre otros:

- Varios idiomas disponibles.
- Diferentes convenciones culturales.
- Zonas horarias.
- Formatos de horarios.
- Formatos de fechas.
- Monedas internacionales.
- Sistema de pesos y medidas (pulgadas/centímetros, libras/gramos…).
- Códigos de caracteres (Unicode resuelve fácilmente este problema).
- Formato de números (separadores decimales, separadores de miles…).

3.- Codificación ISO-639 e ISO-3166

ISO 639:

Clasificación de los **idiomas** en códigos de dos o tres caracteres en minúsculas:

- es → español
- fr → francés
- de → alemán

ISO 3166:

Clasifica los **nombres de los países** en códigos de dos caracteres en mayúsculas:

- ES → España
- AR → Argentina
- UY → Uruguay

3.- Codificación ISO-639 e ISO-3166

Se usan combinaciones de las dos codificaciones ISO-639 e ISO-3166 para indicar el idioma y el país:

en_AU: inglés Australia

es_ES: español España

4.- Idiomas del sistema operativo

Para poder acceder a una aplicación web se utiliza un navegador web que se ejecuta sobre un sistema operativo (SO).

El sistema operativo está configurado con un idioma que generalmente elige el usuario en la instalación del SO.

Al instalar un navegador web, este se configura automáticamente con el idioma del SO del equipo y posteriormente se puede cambiar.

Al desarrollar una aplicación web **localizada** es importante conocer el idioma en el que está configurado en navegador web del usuario.

4.- Idiomas del sistema operativo

En PHP se puede utilizar la variable superglobal **$_SERVER** para conocer (si existe) la lista de idiomas que se aceptan desde el navegador web.

$_SERVER['HTTP_ACCEPT_LANGUAGE']

Contendrá algo parecido a los siguientes ejemplos:

es-ES,es;q=0.9
es-ES,es;q=0.9,en;q=0.8

La **q** indica la prioridad del idioma.

4.- Idiomas del sistema operativo

Dependiendo de la configuración detectada se puede mostrar la información en un idioma u otro.

En aplicaciones web multiidioma se ofrece al usuario la oportunidad de configurar un idioma diferente y siempre que entre en la aplicación se mostrará ese idioma.

Para almacenar el idioma personalizado por un usuario se pueden usar cualquiera de las técnicas aprendidas hasta el momento:

- Cookie
- Sesión
- Base de datos

4.- Idiomas del sistema operativo

Las primeras aproximaciones para realizar la localización de una aplicación web pueden consistir en:

- En la página web **mediante instrucciones if** se muestra un contenido u otro.

- Crear **diferentes versiones de una página web** añadiendo el idioma al nombre → sobrenosotros.esES.php

Práctica

Actividad 1:

Información multi idioma de un personaje de ficción.

7.- Técnicas

Existen diferentes técnicas para abstraer el idioma de la aplicación de la lógica de la misma:

- **Base de datos**: en la base de datos se crea una tabla con una columna para el identificador de la cadena a traducir y además, otras tantas columnas como idiomas tenga la aplicación.
- **Array**: se crean tantos archivos php como idiomas tenga la aplicación, en cada archivo se crea el mismo array asociativo que contiene las cadenas traducidas.
- **CSV**, **JSON**: similar a la técnica del array pero su uso es más complicado al tener que tratar los archivos para extraer la información.
- **Gettext**: biblioteca GNU de internacionalización, requiere un software de edición por lo que se evitan los problemas de usar texto plano. PHP contiene funciones para trabajar con Gettext.

7.- Técnicas

Soluciones **base de datos** y archivos **CSV** o **JSON**.

Estas soluciones tienen las siguientes desventajas:

- Su uso implica demasiada complejidad.
- Se depende del programador para realizar cambios.

Estas desventajas implican que no se aconsejen estas técnicas.

6.- Idiomas en Arrays

Esta técnica consiste en crear un archivo **PHP** por cada idioma en el que se puede localizar la aplicación web.

- **includes/lang/es.inc.php**
- **includes/lang/en.inc.php**
- **includes/lang/fr.inc.php**
- **includes/lang/jp.inc.php**

En cada uno de estos archivos se incluirá **el mismo array** pero cada uno tendrá las cadenas traducidas al idioma indicado en el nombre del archivo.

6.- Idiomas en Arrays

includes/lang/es.inc.php

```php
<?php
$message['titulo'] = 'Mi primera página web';
$message['presentacion'] = 'Bienvenidos a mi web';
```

includes/lang/en.inc.php

```php
<?php
$message['titulo'] = 'My first web site';
$message['presentacion'] = 'Welcome to my site';
```

6.- Idiomas en Arrays

Es muy habitual que las claves usadas en el array se incluya el carácter punto . Para crear un orden en las cadenas:

```php
<?php
$message['header.titulo'] = 'Mi primera página web';
$message['header.presentacion'] = 'Bienvenidos a mi web';
$message['section.titulo'] = 'Nuestros productos';
```

6.- Idiomas en Arrays

Una vez completados todos los archivos con las traducciones, al principio de cada **script PHP** de la aplicación se debe incluir uno de los archivos de idioma.

El archivo de idioma que se incluya puede ser el del idioma por defecto del navegador o bien uno elegido por el usuario y almacenado previamente (base de datos, cookie o sesión).

Una vez detectado el idioma:

```php
require_once('includes/lang/'. $idiomaDetectado .'.inc.php');
```

6.- Idiomas en Arrays

Si el idioma detectado es diferente al idioma por defecto, es habitual incluir primero el idioma por defecto y a continuación el idioma detectado.

Esto se realiza para que si alguna de las cadenas no está traducida se muestre en el idioma por defecto.

Como los elementos del array son los mismos con la segunda inclusión se sobrescriben las cadenas del idioma por defecto.

Por ejemplo, idioma detectado:inglés, idioma por defecto: español:

```
require_once('includes/lang/es.inc.php');
require_once('includes/lang/'. $idiomaDetectado .'.inc.php');
```

6.- Idiomas en Arrays

A continuación, se debe utilizar el array para tener localizada la aplicación.

```
<!doctype html>
<html>
   <head>
      <meta charset="utf-8">
      <title><?=$message['titulo']?></title>
   </head>

   <body>
      <?=$message['presentacion']?>
   </body>
</html>
```

6.- Idiomas en Arrays

Cadenas con parámetros

Hay ocasiones en las que interesa que las cadenas que se traducen contengan parámetros que cambian según el momento de ejecución del script.

Por ejemplo: Hay 5 productos en el carrito.

Para realizar esta funcionalidad se utilizan las cadenas con formato y la función printf en lugar de echo.

6.- Idiomas en Arrays

Cadenas con parámetros

En el archivo de idioma:

```
$message['cantidadProductos'] = 'Hay %u productos en el carrito.';
```

Cuando se necesita usar la cadena:

```
printf($message['cantidadProductos'], $cantidadProductos);
```

u → entero sin signo.
b → entero representado como binario.
f → decimal (considerando configuración regional: punto o coma).
s → string.

6.- Idiomas en Arrays

La técnica de los array se utiliza mucho cuando la aplicación web no tiene excesivos idiomas y no hay muchas cadenas de texto que traducir.

A más grande sea el proyecto, más archivos de localización y más cadenas contendrá para traducir.

Práctica

Actividad 2:

MerchaShop internacional.

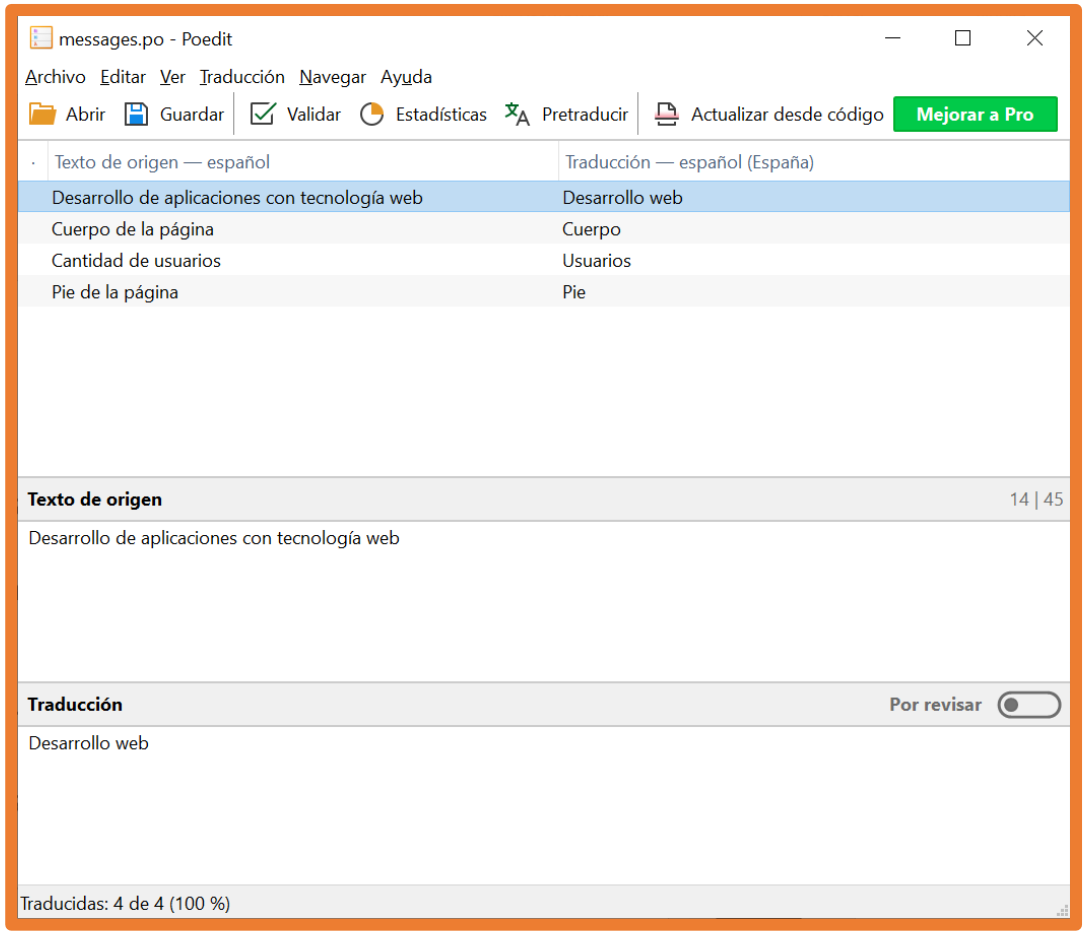
7.- Idiomas con Gettext

La principal ventaja del uso de Gettext es que se puede usar un software externo para las traducciones.

De esta manera no es necesario que los desarrolladores no se centren en la traducción y esta tarea recaiga en gente especializada en la localización

Algunos ejemplos de software:

Poeditor y Poedit



8.- Creación de cadenas a traducir – Gettext

En el código fuente de la aplicación web se deben marcar las cadenas a traducir con la función **gettext()** o su alias **_()**.

Al marcar las cadenas el editor de traducciones será capaz de generar automáticamente los archivos de traducción.

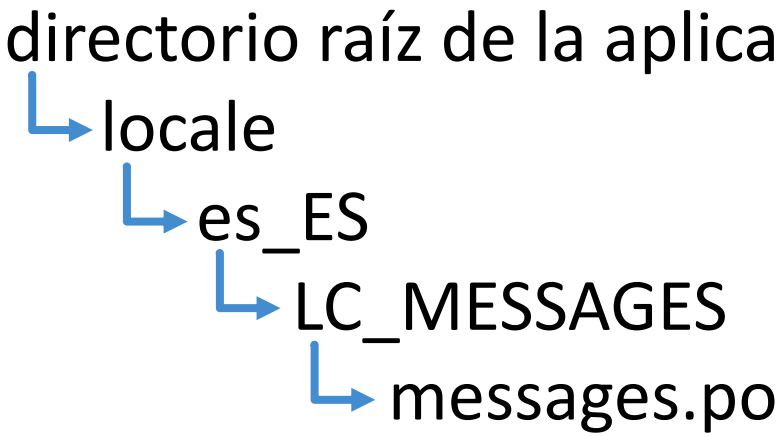
```
echo gettext('Bienvenidos a mi pagina web');
echo _('Lista de productos');
```

Si no existen archivos de traducción se mostrará directamente las cadenas por defecto, que son las usadas directamente en gettext() o _().

El inconveniente es que las cadenas en gettext() o _() no pueden tener caracteres especiales como acentos, por eso se suelen usar el inglés para estas cadenas y generar archivos de traducción para otros idiomas.

9.- Archivos con las traducciones – Gettext

Los archivos de traducción tienen la extensión **.po** y deben guardarse con la siguiente estructura de directorios:

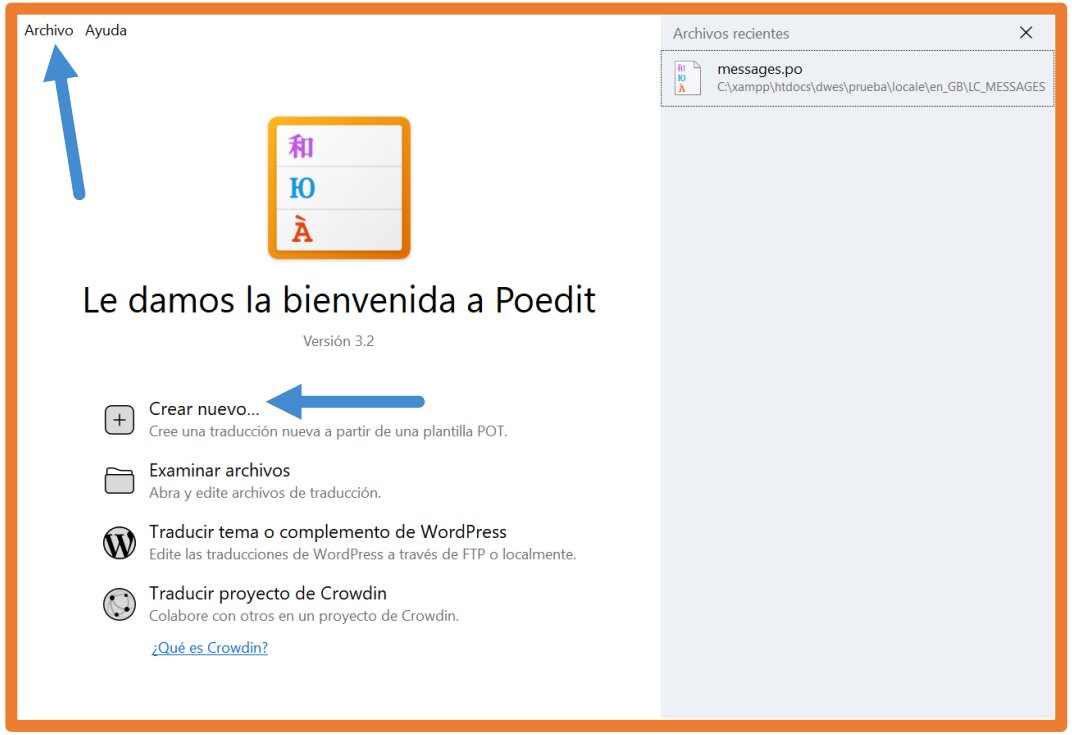


Dentro del directorio **locale** puede haber tantos directorios como localizaciones con traducciones se quiera.

9.- Archivos con las traducciones – Gettext

Una vez marcadas todas las cadenas que se van a traducir en el código fuente, se debe proceder a crear los archivos de traducción mediante el software indicado.

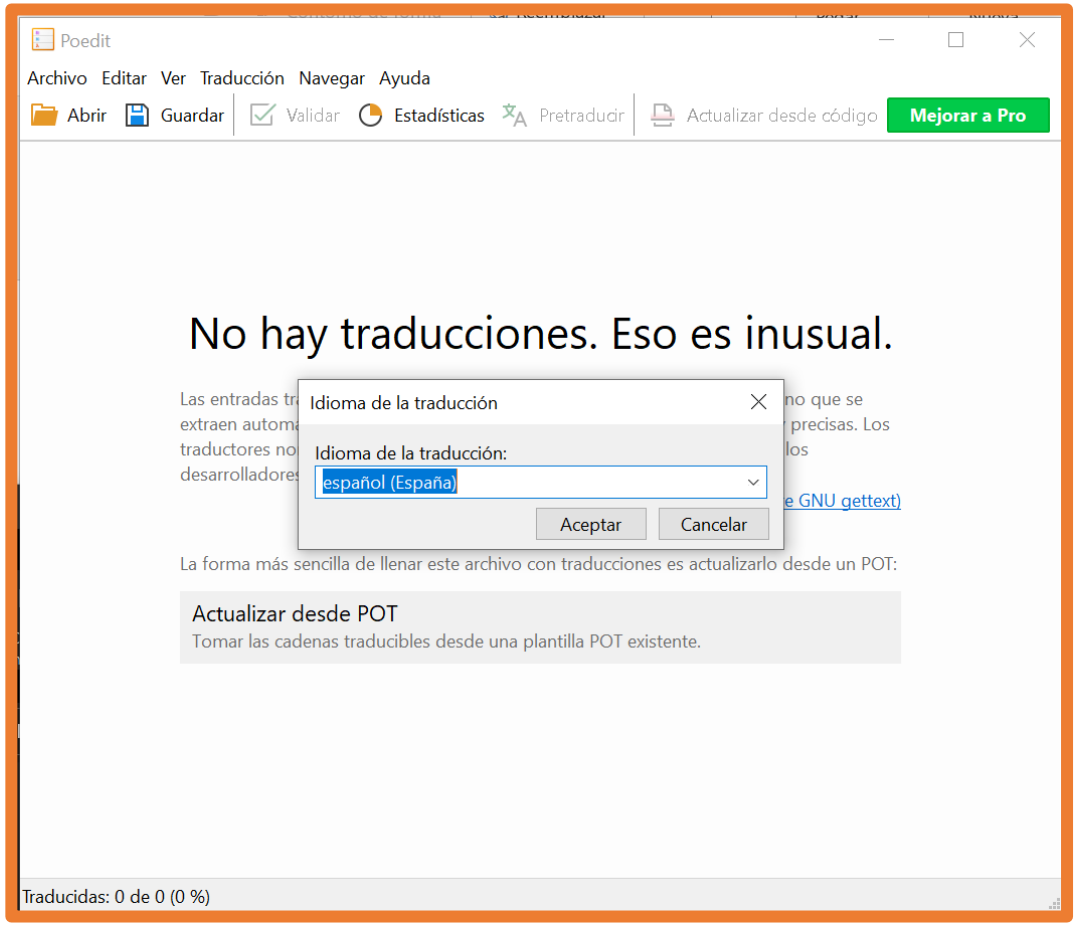
Por ejemplo en Poedit se debe crear un archivo nuevo: Archivo → Nuevo…



9.- Archivos con las traducciones – Gettext

Se debe elegir el idioma para el que se quiera crear la traducción.

Tras crear el archivo lo primero que debe hacerse es guardarlo.

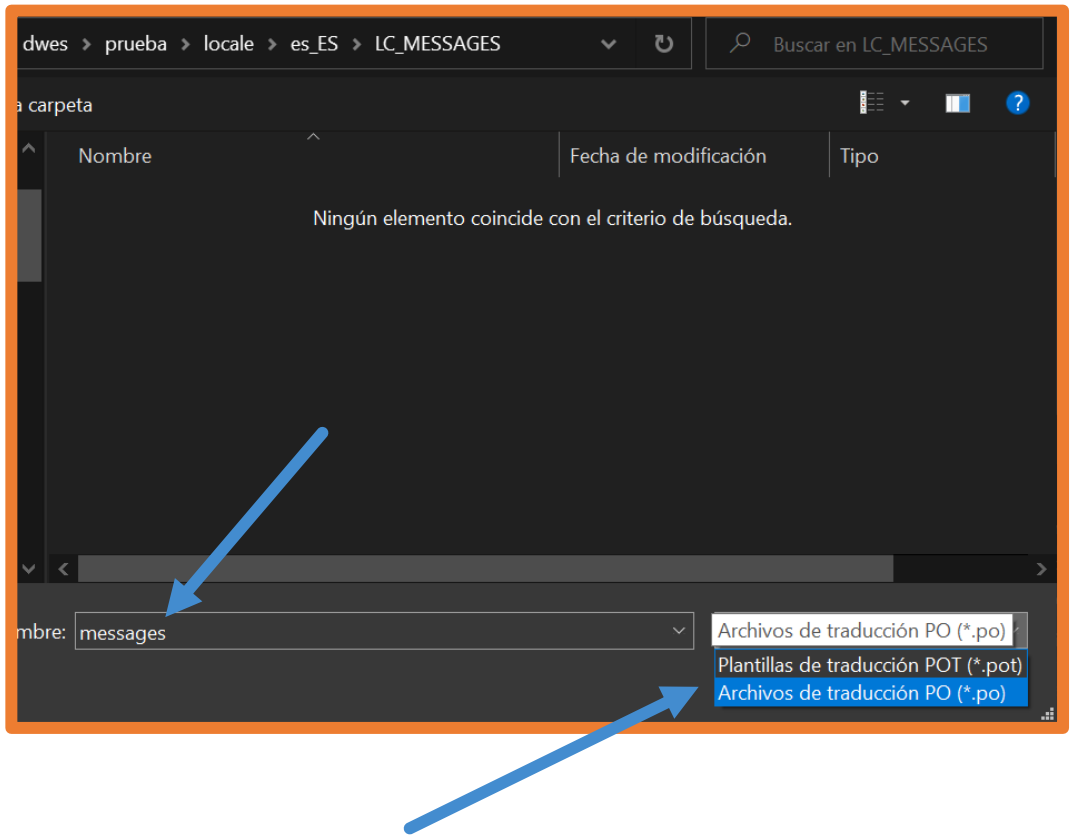


9.- Archivos con las traducciones – Gettext

Se debe guardar el archivo en el directorio correcto y asegurarse que la extensión es .po.

El nombre del archivo puede ser cualquiera, los nombres típicos son:

messages.po
lang.po



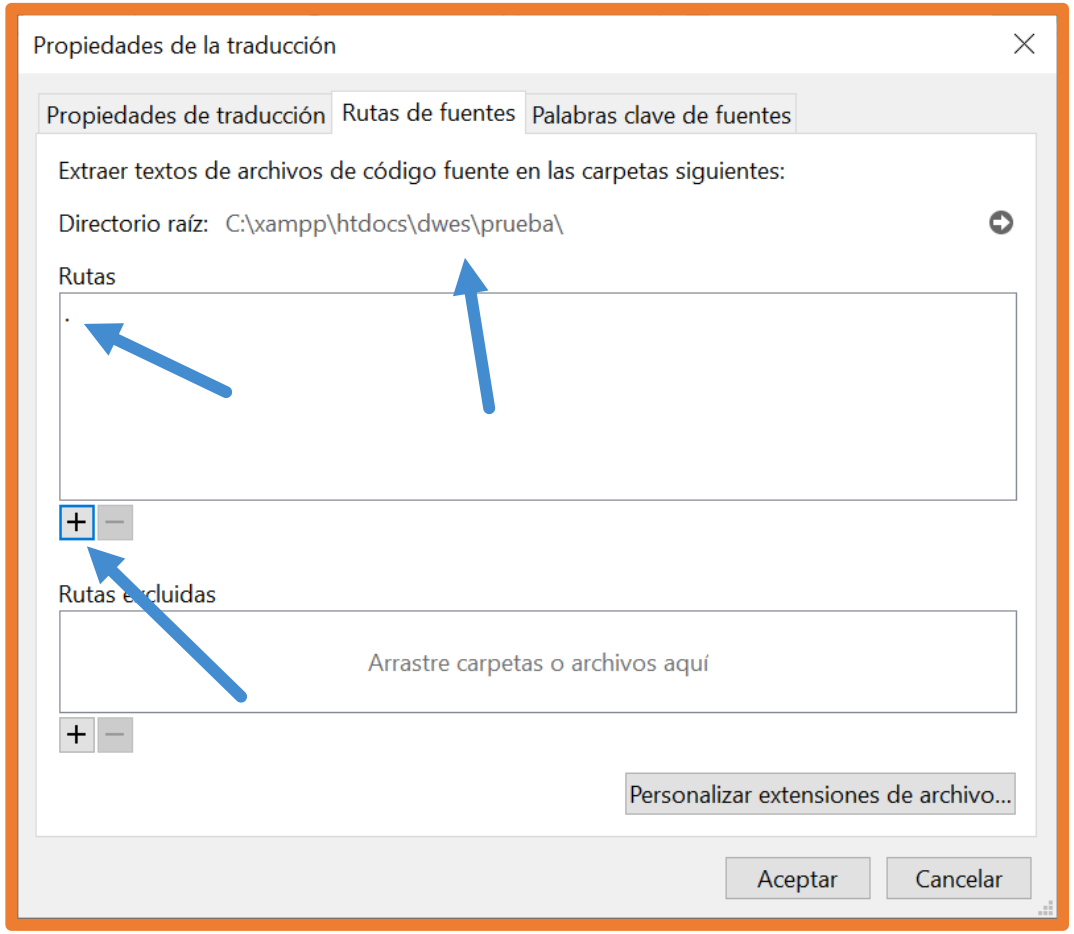
9.- Archivos con las traducciones – Gettext

El siguiente paso es indicarle al programa en qué carpeta debe escanear los archivos para buscar cadenas traducibles.

Traducción→ Propiedades

Pestaña: Rutas de fuentes

En el botón "+" se debe añadir el directorio raíz de la aplicación web.

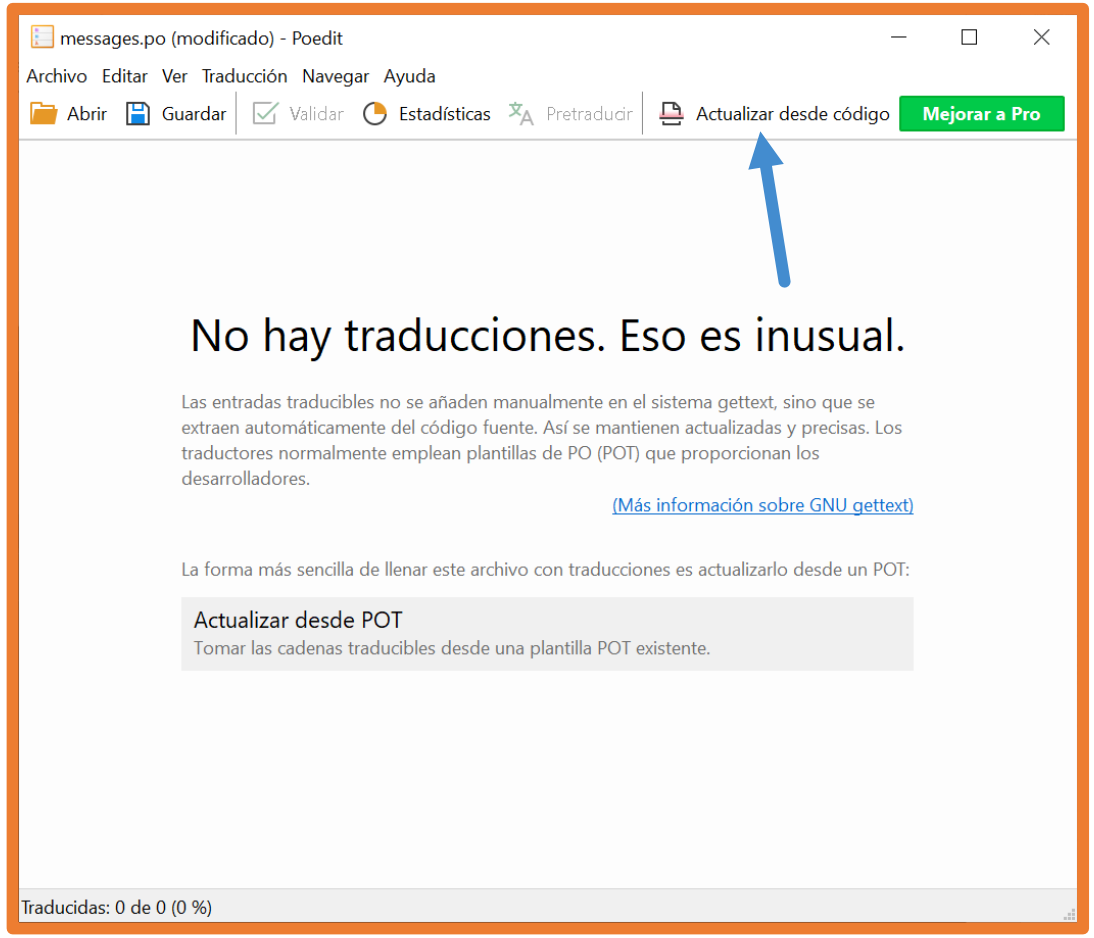


9.- Archivos con las traducciones – Gettext

Una vez seleccionado el directorio se tendrá que pulsar la opción:

"Actualizar desde código"

De esta manera Poedit reconocerá todas las cadenas y las añadirá al archivo.



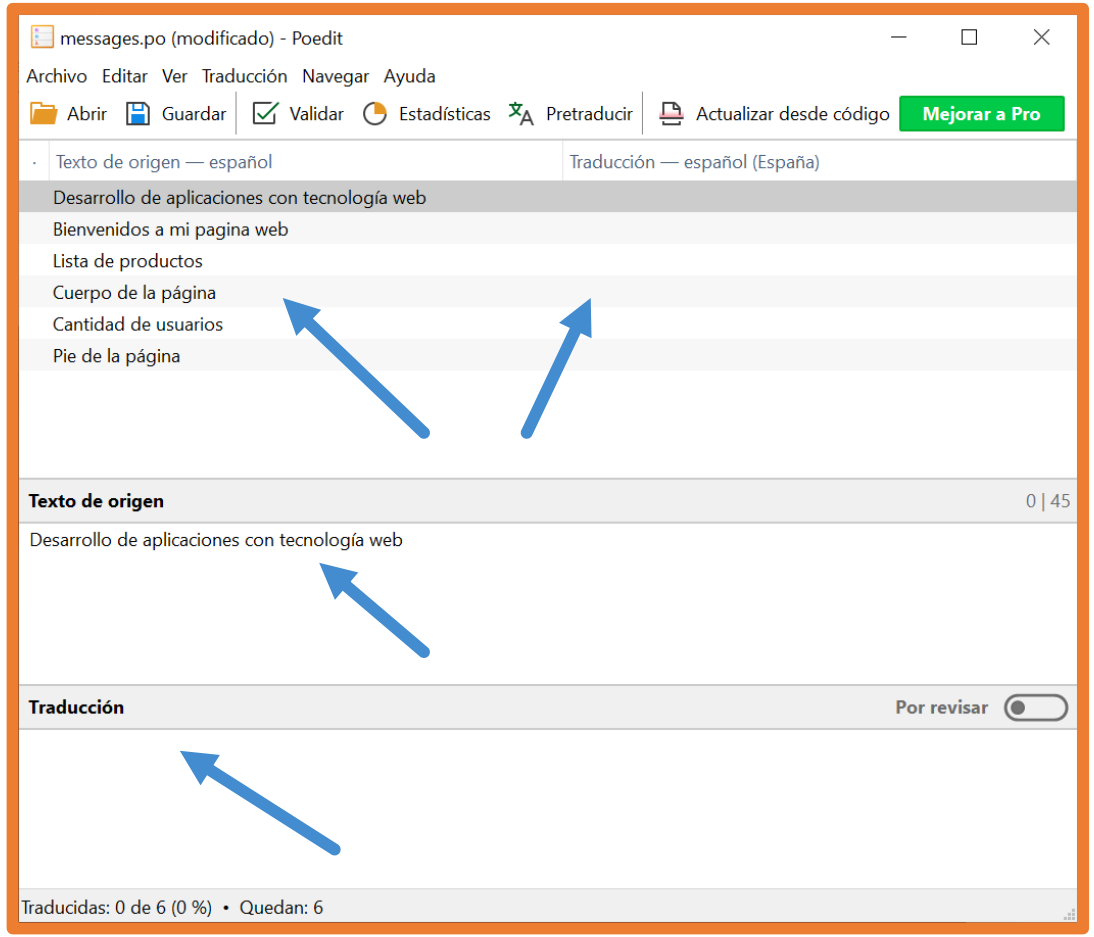
9.- Archivos con las traducciones – Gettext

Una vez seleccionado el directorio se tendrá que pulsar la opción:

"Actualizar desde código"

De esta manera Poedit reconocerá todas las cadenas y las añadirá al archivo.

Este es el momento de crear las traducciones.

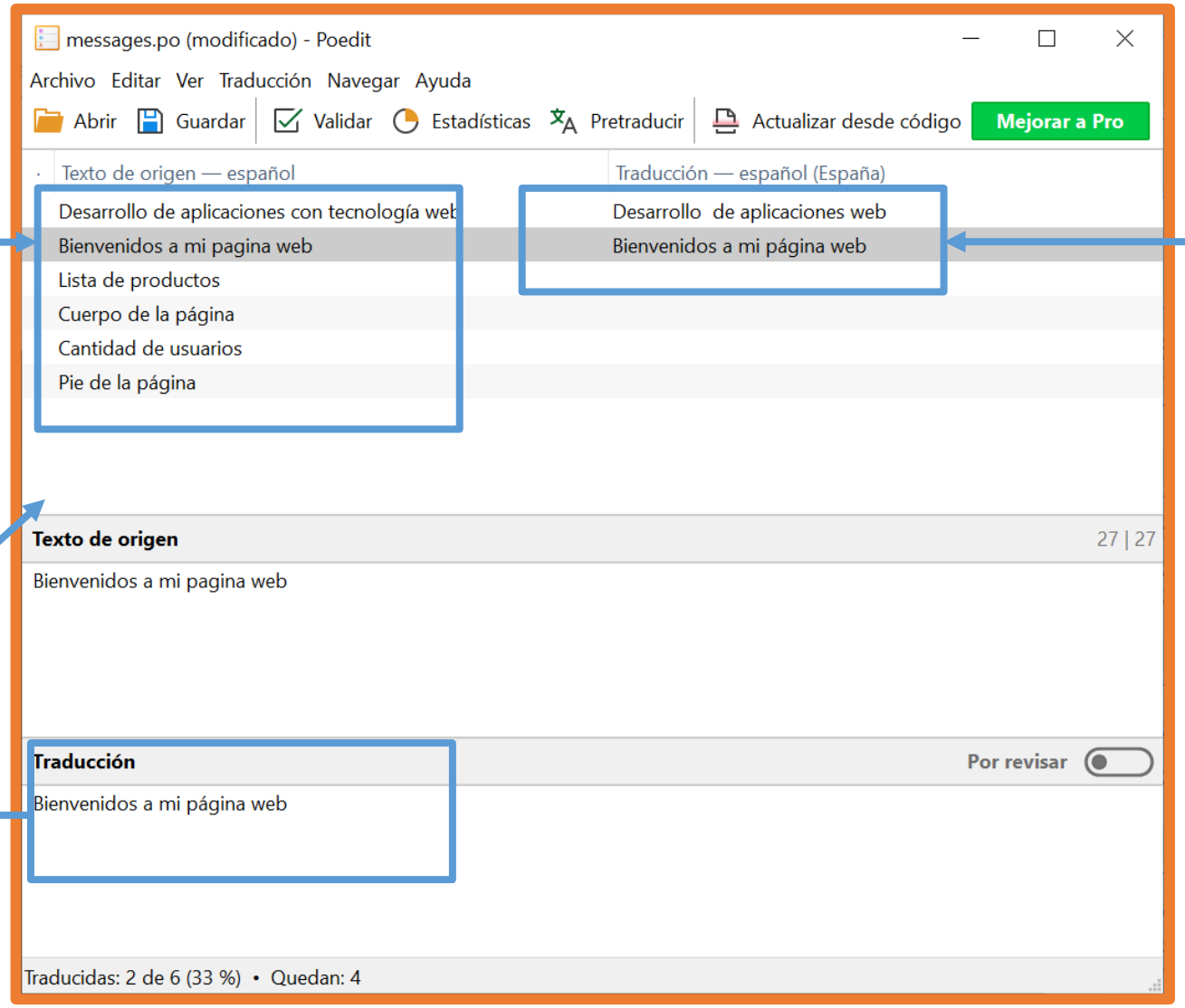


9.- Archivos con las traducciones – Gettext

Cadenas a traducir

Traducción de la cadena seleccionada

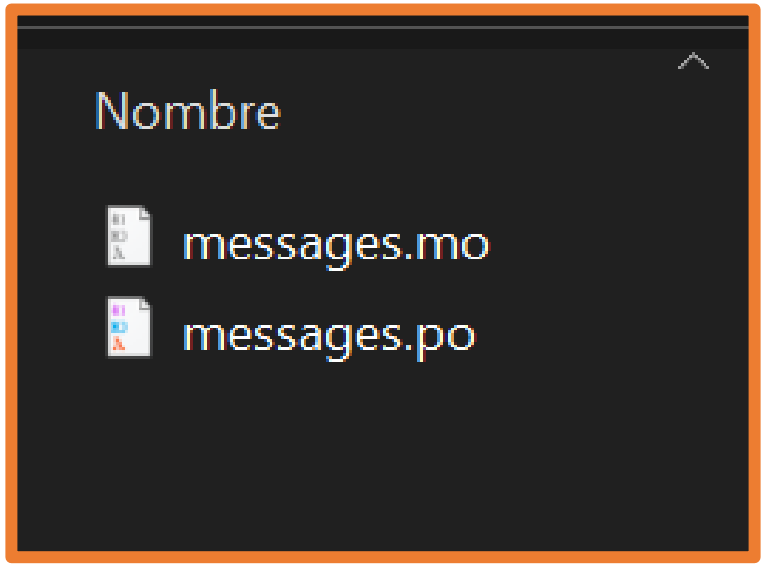
Traducciones



9.- Archivos con las traducciones – Gettext

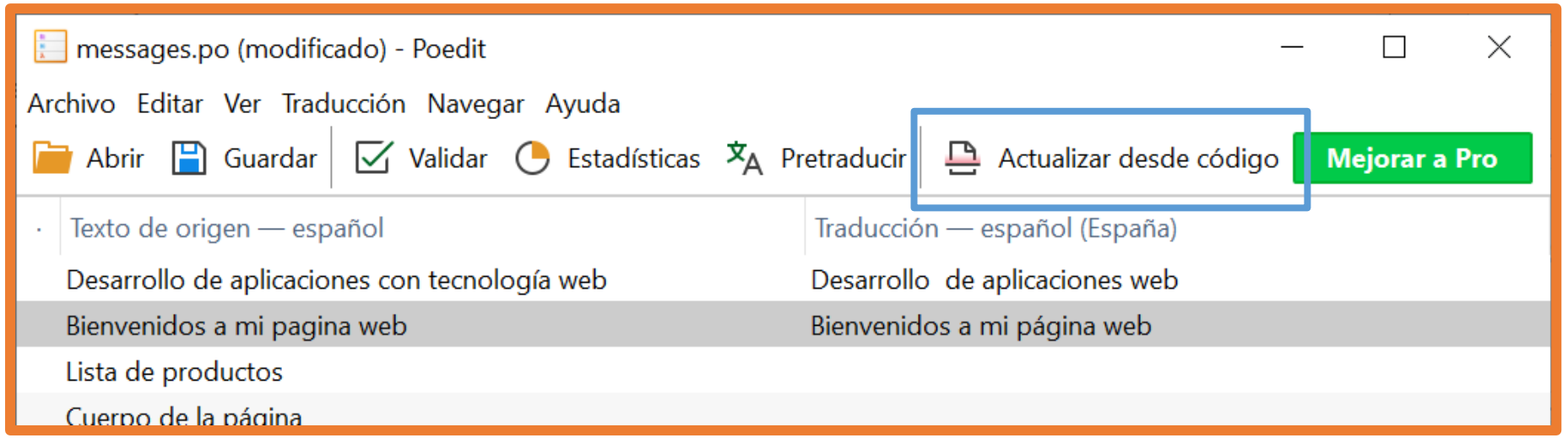
Una vez traducidas todas las cadenas se debe guardar.

Al guardar se generará un archivo nuevo con extensión **.mo** que contiene las cadenas traducidas en formato binario para ser utilizadas directamente desde el intérprete de PHP.



9.- Archivos con las traducciones – Gettext

Si más adelante se añaden nuevas cadenas a traducir a la aplicación web, en Poedit se tendrá que abrir el archivo po y volver a pulsar el botón "Actualizar desde código" para que se reconozcan las cadenas nuevas.



10.- Uso de las traducciones – Gettext

Para indicarle a PHP que seleccione el archivo correcto de traducciones se debe obtener el idioma del navegador o el elegido por el usuario y al principio de cada script se debe ejecutar el siguiente código:

```php
$idioma = "es_ES";
setlocale(LC_ALL, $idioma); // para servidor en linux
putenv("LC_ALL=". $idioma); // para servidor en Windows
bindtextdomain("messages", 'locale');
bind_textdomain_codeset("messages", 'UTF-8');
textdomain("messages");
```

10.- Uso de las traducciones – Gettext

La selección de idioma funciona en base a los **locale** que haya instalados en el sistema operativo del servidor web.

En Linux con el comando **locale -a** se pueden consultar los **locale** instalados.

Modificando el archivo **etc/locale-gen** se pueden añadir nuevos **locale**.

Tras añadir un **locale** se debe reiniciar el servidor web.

En Windows los **locale** suelen dar problemas, pero no es habitual que un servidor Apache en internet trabaje bajo Windows.

11.- Conclusiones

Según el alcance que se desee para una aplicación web es importante que esta se encuentre internacionalizada y localizada.

Con la velocidad actual de los procesadores no es un factor importante elegir una técnica u otra.

Las técnicas más usadas son los archivos po y los arrays porque es fácil que una persona no programadora traduzca las cadenas de la aplicación.

El framework **Laravel** utiliza la traducción mediante arrays.