

# Internacionalización

---

INTERNACIONALIZACION Y LOCALIZACION EN JAVA

# Introducción

---

- El objetivo de la internacionalización es hacer llegar un producto a mercados internacionales.
- Los usuarios de dichos productos difieren entre sí en muchos aspectos:
  - Idioma.
  - Aspectos culturales.
  - Otros: formatos, símbolos...



# Internacionalización (i18n)

---

- Proceso de construir un producto para que sea utilizado en el mercado internacional, especificándolo en el diseño de sus características y en el código.
- Diseño y desarrollo de un producto, aplicación o documento que permite una **fácil localización** para las diversas culturas, regiones o lenguaje de las audiencias destino
- Es el proceso de diseñar una aplicación de tal manera que pueda adaptarse a diferentes lenguajes y regiones sin necesidad de cambiar el código.
  - Un programa internacionalizado no tiene elementos dependientes de la lengua o del contexto cultural de un país o región en el propio código.
  - Los elementos textuales, como los mensajes o las etiquetas de los componentes, no están en el código y se incorporan de manera dinámica.

# Internacionalización: Ventajas

---

- Tradicionalmente, el desarrollo de software internacional se ha hecho en dos etapas:
  - Inicialmente se desarrollaba la aplicación en un idioma determinado
  - Posteriormente se duplicaba el código de la aplicación para traducirlo y adaptarlo. Al final había tantos códigos de la aplicación como adaptaciones a diferentes países se hubiesen realizado
- Sin embargo, si se considera desde el diseño de la aplicación la posibilidad de que ésta tenga que adaptarse a diferentes mercados las **ventajas** que se obtienen son varias.
  - Se pueden crear versiones localizadas más fácil y rápidamente.
  - Con la adición de datos de localización, el mismo ejecutable puede ser ejecutado en cualquier lugar del mundo
  - El soporte de nuevos idiomas no requiere re-compilación
  - Se utilizan los recursos más eficientemente
  - El mantenimiento del código y la inclusión de nuevas localizaciones es menos costoso ya que sólo hay una versión del mismo
  - El mercado es mayor

# Ventajas de la internacionalización (II)

---

“Desarrollar una web en un idioma y traducirla después, es mucho más caro y costoso (hasta el doble) que desarrollarla en varios idiomas desde su concepción”



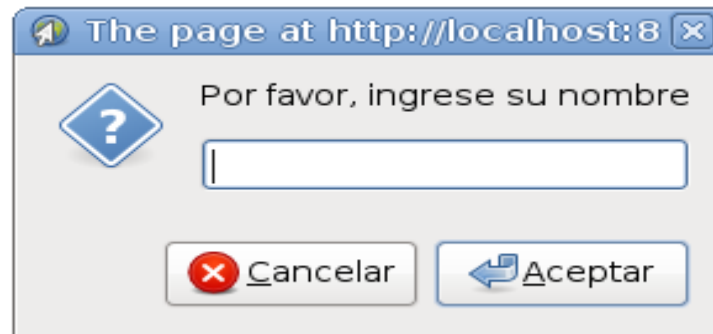
Richard Ishida. Ex-director de la Actividad de Internacionalización del W3C (<https://www.w3.org/International/>)

<https://www.w3.org/groups/wg/i18n-core>

# Localización (l10n)

---

- Adaptación de un producto software al lenguaje y a las particularidades regionales, de cultura o lenguaje de la audiencia destino.
- Cada país representa una localización única: un idioma puede ser hablado en más de un país como, por ejemplo, el español, pero hay diferencias como la moneda, entre otras cosas. Por esto, cada país representa una localización única.



# Localización (II)

En Windows se puede especificar una localización concreta en el *Panel de Control*

- *Región y Reloj – Región – Preferencias de Idioma*



# Elementos Específicos de la Localización

---

- A la hora de internacionalizar una aplicación hay que tener en cuenta:
  - Los mensajes y texto que aparecen en una aplicación y que deben ser traducidos
  - Y además, los elementos que son específicos de una localización
    - Calendarios, formatos de fechas, horas, etc.
    - Formatos de números y monedas
    - Unidades de medida
    - Iconos, colores
    - Honores y títulos
    - Direcciones ...



# Elementos Específicos de la Localización (II)

---

## **Calendarios.** Diferentes estilos de calendarios:

- Gregoriano, budista, chino, hebreo.
- En el gregoriano se utilizan diferentes formatos:
  - aa/mm/dd
  - aaaa/mm/dd
  - dd/mm/aa
  - mm/dd/aa
- Ejemplos:
  - 23 de abril de 1985      ----- España
  - 4/23/85                      ----- Reino Unido
  - 23/4/85                      ----- Dinamarca
  - 1985-04-23                ----- Suecia

# Elementos Específicos de la Localización (III)

---

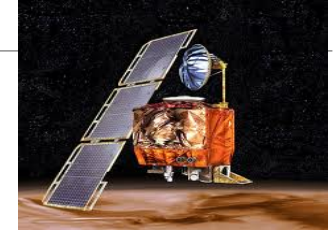
## Tiempo Horario:

- USA ----- 8:32
- Canadá ----- 20:32
- Suiza ----- 20,32,00
- Alemania ----- 20.32 Uhr
- Noruega ----- KI 20.32

## Formatos Monetarios:

- USA ----- \$1,234.56
- Noruega ----- Kr1.234,56
- Suiza ----- sFr1234,56
- Alemania ----- 1.234,56DM

# Elementos Específicos de la Localización (IV)



## Números y Símbolos numéricos

- EEUU
  - Billón, uno seguido de nueve ceros
  - La coma separa miles
- América Latina y Europa
  - Billón, uno seguido de doce ceros
  - La coma separa decimales

## Unidades de medida

- La mayoría sistema métrico decimal (Km, cm,...)
- EEUU emplea la milla y la pulgada

## Temperatura

- Celsius / Fahrenheit

## Pérdida de la Mars Climate Observer

Uno de los fracasos más sonados en la conquista de Marte fue el ocurrido con la "Mars Climate Observer", una sonda enviada a Marte por la NASA cuya finalidad era el estudio del clima atmosférico en Marte. Este error costó la nada despreciable cifra de 125 millones de dólares.

Causa de error:

El Jet Propulsion Laboratory de Pasadena, encargado de programar los sistemas de navegación de la sonda, utilizó el sistema métrico decimal (milímetros, metros, kilómetros y kilos) para realizar sus cálculos, mientras que otro laboratorio, el Lockheed Martin Astronautics de Denver, que diseñó y construyó la Mars Climate Observer, utilizó el sistema inglés (pulgadas, pies y libras). Sin embargo, los datos de navegación no fueron convertidos de un sistema a otro antes del lanzamiento al espacio de la Mars Climate.

[http://elpais.com/diario/1999/10/02/sociedad/938815207\\_850215.html](http://elpais.com/diario/1999/10/02/sociedad/938815207_850215.html)

# Elementos Específicos de la Localización (V)

## Tamaños de papel

- Mayoría: DIN A-3, DIN A-4
- EEUU: legal, letter, ledger

## Formato direcciones postales

- Mayoría: nombre de la calle + número
- EEUU: número + nombre de la calle

## Signos de puntuación

- Algunos símbolos no utilizados: ¿, ¡,...
- Griegos: signo de interrogación parecido al ;

## Colores

- Occidentales: rojo parar, verde continuar. Convención no válida en China
- En U.S.A. el color blanco significa pureza, en Japón significa muerte. En Egipto, el color rojo representa la muerte, en China sugiere felicidad.

## Iconos

- ✋, 🖐️: Ofensivos en Grecia, Brasil
- ✓: Correcto en varios países; en otros, como Japón, significa Incorrecto



# Elementos Específicos de la Localización (VI)

---

Culturas con lenguajes que se escriben de derecha a izquierda



Copyright © 2000—2003 Tex Texin, Stéphanie Smith. All Rights Reserved

# Además: Aspectos lingüísticos (I)

---

Entre diferentes localizaciones se debe tener en cuenta que:

- 1. Una palabra puede expanderse
  - TRIM – EFFACEMENTS DE BORDS – ELIMINACIÓN DE BORDES
  - INPUT PROCESSING FEATURES – EINGABEVERARBEITUNGSFUNKTIONEN
- 2. En mensajes compuestos, puede ser necesario intercambiar variables
  - There are 268 validation errors in the file Hola.html  
Die Datei Hola.html enthält 268 Gültigkeitsfehler

# Además: Aspectos lingüísticos (II)

- 3. Puede variar el orden de los elementos de navegación o de las opciones propuestas: Ordenación importante en el proceso de internacionalización



[www.yahoo.es](http://www.yahoo.es)



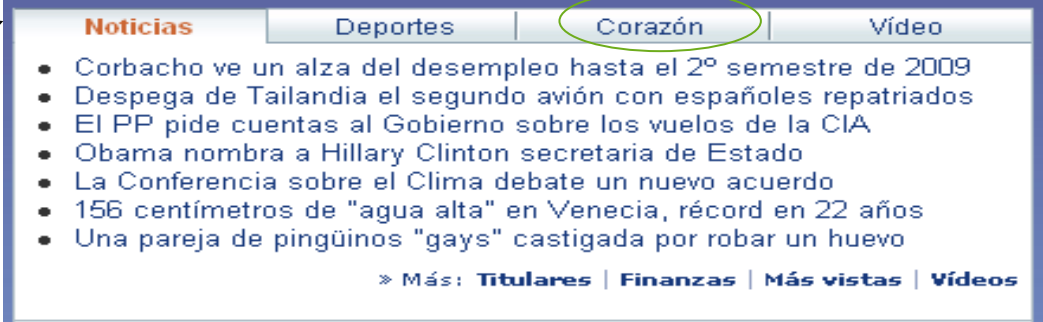
[it.yahoo.com](http://it.yahoo.com)

# Además: Aspectos culturales

## Yahoo España

- Quizás cada cultura priorice de manera diferente sus intereses
- Contenido que pueda ser malinterpretado o considerado ofensivo
- Diferencia en aspectos/exigencias legales

...



The screenshot shows the Yahoo España homepage with the 'Corazón' tab selected. The main content area lists several news items:

- Corbacho ve un alza del desempleo hasta el 2º semestre de 2009
- Despega de Tailandia el segundo avión con españoles repatriados
- El PP pide cuentas al Gobierno sobre los vuelos de la CIA
- Obama nombra a Hillary Clinton secretaria de Estado
- La Conferencia sobre el Clima debate un nuevo acuerdo
- 156 centímetros de "agua alta" en Venecia, récord en 22 años
- Una pareja de pingüinos "gays" castigada por robar un huevo

At the bottom, there are links: » Más: **Titulares** | **Finanzas** | **Más vistas** | **Videos**

## Yahoo Italia



The screenshot shows the Yahoo Italia homepage with the 'Economia' tab selected. The main content area lists several news items:

- Guatemala, sparatoria tra bande rivali a corsa cavalli: 17 morti
- Iraq, Obama: ritiro truppe Usa possibile in 16 mesi
- Obama promette di investire ancora in apparato militare
- Iraq, attentati tra Baghdad e Mosul, 30 morti
- Brasile lancia piano per ridurre deforestazione Amazzonia a 70%
- Canzoniere Mitico, top libri musica
- Svaligiata auto Gabriel Garko

At the bottom, there are links: » Altro: **Video** | **Ultime notizie** | **Fotogallery** | **Gossip**



# Proceso de Internacionalización

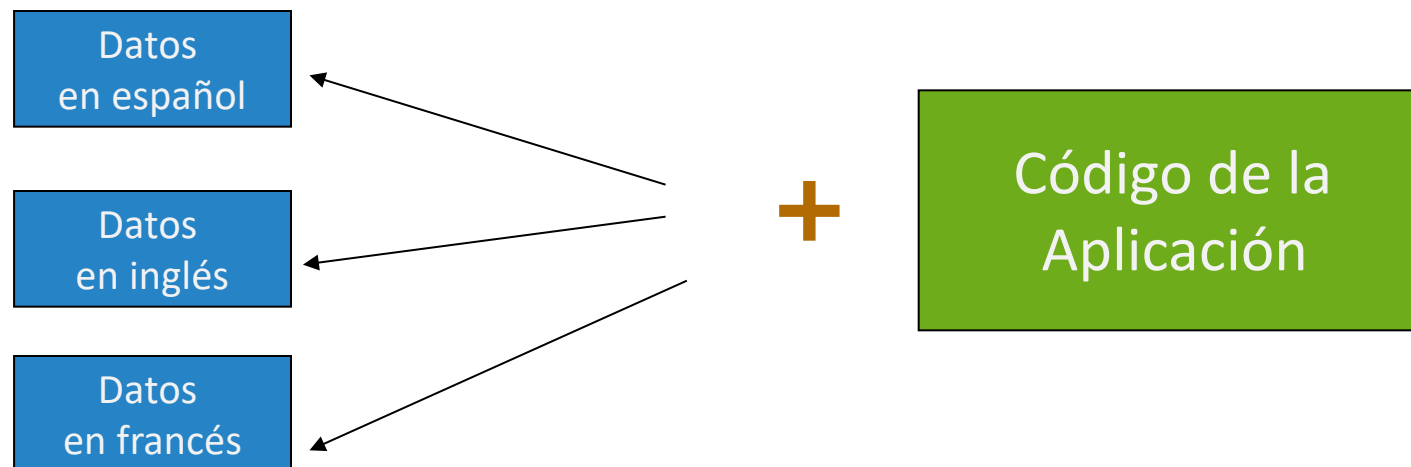
---

- A la hora de internacionalizar una aplicación la idea perseguida es básicamente separar en la aplicación el código de los datos:
  - El **código** hace referencia a la lógica de la aplicación, es decir, el código que se ejecuta igualmente para todos los países y lenguas.
  - Los **datos** hacen referencia a todos aquellos elementos (cadenas, mensajes, iconos, etc.) que cambian de un país a otro (es decir, de una localización a otra).

# Proceso de Internacionalización (II)

---

- La mejor forma de proporcionar contenido internacionalizado es separar del código toda la información sensible a localización.
- La idea se basa en tener :
  - Un **único bloque de código**, es decir, el código de la aplicación siempre va a ser único y el mismo. En este código se considerarán los cambios de formatos de fechas, monedas...
  - Y **varios bloques de datos**, uno por cada país (localización)



# Internacionalización en Java

---

- Se deben tener en cuenta dos tipos de contenido, ya que se van a localizar de manera diferente:
  - **Fechas, números, monedas:** se formatean (mediante las instrucciones adecuadas en el código) de acuerdo a la localización.
  - **Textos:** se recuperan de un fichero de recursos (“Datos en español”, “Datos en inglés”, “Datos en francés”)

# Internacionalización en Java (II)

---

- Clases principales implicadas en la internacionalización en Java (java.util.\*)
  - Locale
  - ResourceBundle
  - NumberFormat
  - DateFormat

# Clases Java para Internacionalización (I)

---

- Clase Locale

- Identifica un idioma y un país concreto, es decir, una localización.

- en US -> inglés Estados Unidos
- en UK -> inglés Reino Unido
- es ES -> español España
- Todos estos códigos están disponibles en:
  - [http://www.loc.gov/standards/iso639-2/php/code\\_list.php](http://www.loc.gov/standards/iso639-2/php/code_list.php) (ISO 639-2)
  - [http://www.iso.org/iso/country\\_codes/iso-3166-1\\_decoding\\_table.htm](http://www.iso.org/iso/country_codes/iso-3166-1_decoding_table.htm) (ISO 3166-1)

- Clase ResourceBundle

- Contiene los objetos específicos a la localización

- Clase NumberFormat

- Permite dar formato a valores numéricos en función de la localización.

- Clase DateFormat

- Permite dar formato a fechas y horas en función de la localización.

# Ejemplo 1

---

- Partimos de una clase Saludo.java que muestra tres mensajes por pantalla.
- Queremos que los mensajes aparezcan, dependiendo de la configuración regional del ordenador en:
  - Español
  - Inglés
  - Francés

# Saludo.java

---

```
public class Saludo{  
    public static void main(String [] args){  
        System.out.println("Hola a todos");  
        System.out.println("Escuela de Ingeniería Informática");  
        System.out.println("Universidad de Oviedo");  
    }  
}
```

# Pasos a seguir

---

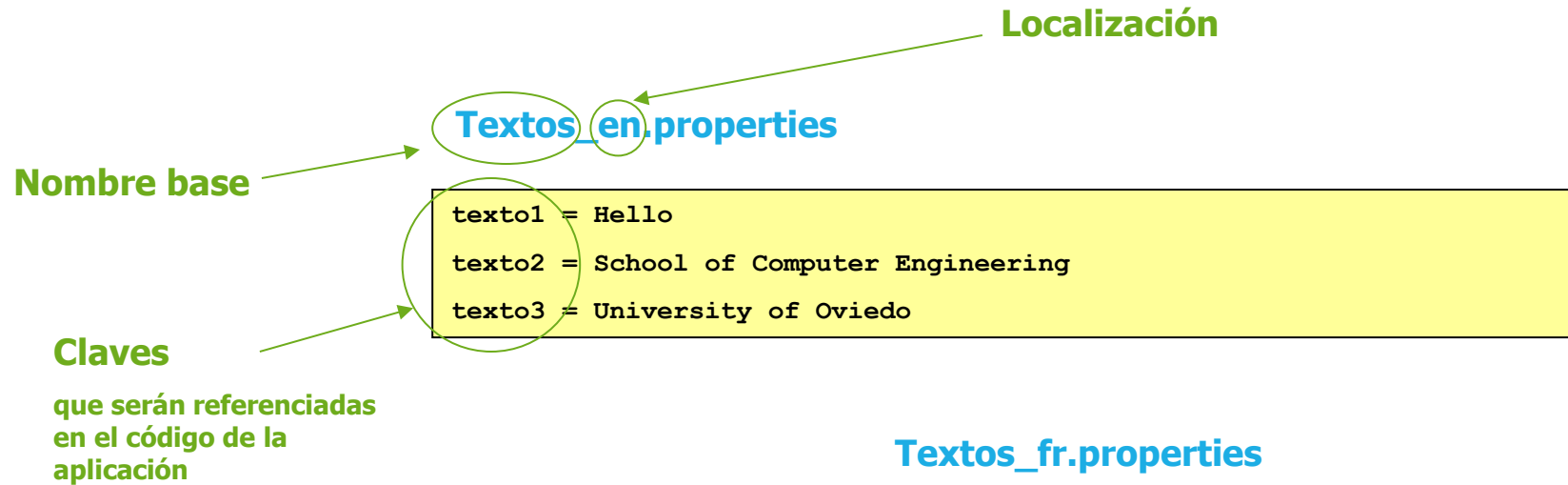
1. Identificar los elementos a localizar, es decir, identificar el bloque de datos. En este caso son los tres mensajes:
  - *Hola*
  - *Escuela de Ingeniería Informática*
  - *Universidad de Oviedo*
2. Crear los ficheros de propiedades.
  - Se trata de ficheros de texto
  - Habrá un fichero para cada localización
  - El fichero tendrá un nombre especial que identifica la localización
    - NombreFichero\_es.properties ( para localización Español)
    - NombreFichero\_en.properties ( para localización Inglés)
    - NombreFichero\_fr.properties (para localización Francés)
    - ...
  - Este fichero sólo puede contener cadenas (Strings) y a cada una de las cadenas se les asignará un nombre (el que queramos)



# Creación de los Ficheros de recursos

## Textos\_es.properties

```
texto1 = Hola  
texto2 = Escuela de Ingeniería Informática  
texto3 = Universidad de Oviedo
```



## Textos\_fr.properties

```
texto1 = Bonjour  
texto2 = École de Ingénierie Informatique  
texto3 = Université d'Oviedo
```

# Pasos a seguir (II)

---

3. Internacionalizar la clase, es decir, adaptarla para que seleccione los datos del fichero adecuado en función de la localización que esté seleccionada en el ordenador. Para ello:

3.1 Obtener la localización seleccionada en el ordenador:

```
Locale localizacion = Locale.getDefault(Locale.Category.FORMAT)
```

Si hubiera que establecer una localización concreta (español, por ejemplo):

```
Locale localizacion = new Locale("es")
```

3.2 Cargar los datos del fichero de propiedades(IIamado Textos) correspondiente, en función de la localización obtenida en el paso anterior:

```
ResourceBundle mensajes = ResourceBundle.getBundle("Textos", localizacion);
```

3.3 Extraer cada uno de los mensajes en función del nombre que se les asignó en el fichero de propiedades:

```
System.out.println(mensajes.getString("texto1"));
```

```
System.out.println(mensajes.getString("texto2"));
```

```
System.out.println(mensajes.getString("texto3"));
```

# Clase Internacionalizada

---

- La clase ya internacionalizada tendría este aspecto:

```
import java.util.*;

public class Saludo2 {

    public static void main(String [] args){

        Locale localizacion = Locale.getDefault(Locale.Category.FORMAT);

        ResourceBundle mensajes = ResourceBundle.getBundle("Textos", localizacion);

        System.out.println(mensajes.getString("texto1"));

        System.out.println(mensajes.getString("texto2"));

        System.out.println(mensajes.getString("texto3"));

    }

}
```

# Nuevas localizaciones

---

- Si ahora deseáramos que nuestra aplicación respondiese a una nueva localización, como por ejemplo italiano, únicamente sería necesario crear un nuevo fichero de propiedades de nombre `Textos_it.properties` con las cadenas traducidas a italiano.
- La clase Saludo2 no sería necesario cambiarla.

# Ficheros de recursos: Otro Ejemplo

## Labels\_es.properties

```
# -- Site -----
site.title = Evaluacion de la Usabilidad OnLine
site.slogan = Evaluador gratuito
site.contact = Contacto
site.contact.title = Correo de contacto
site.home = Inicio
site.home.title = Ir a la pagina de inicio

# -- Menus -----
menu.registeruser = Registrar usuario
menu.registeruser.title = Registrar usuario evaluador
menu.registersite = Registrar sitio
menu.registersite.title = Registre su sitio para ser evaluado

# -- Generic messages -----
day = Día
month = Mes
year = Año
back = Atrás

# -- Images (alt) -----
home.alt = Casa
mail.alt = Sobre con sello
help.alt = Un libro con una interrogacion
password.alt = Una llave
print.alt = Una impresora
```

## Labels\_en.properties

```
# -- Site -----
site.title = Usability Evaluation OnLine
site.slogan = Free evaluator
site.contact = Contact
site.contact.title = Email contact
site.home = Home
site.home.title = Go to the home page

# -- Menus -----
menu.registeruser = Register user
menu.registeruser.title = Register user evaluator
menu.registersite = Register site
menu.registersite.title = Register your site to be evaluated

# -- Generic messages -----
day = Day
month = Month
year = Year
back = Back

# -- Images (alt) -----
home.alt = home
mail.alt = A card with stamp
help.alt = A book with a question mark
password.alt = A key
print.alt = A printer
```

# Ejemplo 2

---

```
import java.util.*;

import java.text.*;

public class FechaHoraNumero {

    public static void main(String [] args){

        Locale localizacion = Locale.getDefault(Locale.Category.FORMAT)

        Date fechaHora = new Date();

        DateFormat formatoFecha = DateFormat.getDateInstance(DateFormat.LONG,localizacion);

        DateFormat formatoHora = DateFormat.getTimeInstance(DateFormat.LONG,localizacion);

        System.out.println(formatoFecha.format(fechaHora));

        System.out.println(formatoHora.format(fechaHora));

        System.out.println(NumberFormat.getNumberInstance().format(123456.78));

    }

}
```