Joaquín Seoane, DIT/UPM

Estas transparencias se ven perfectamente con Mozilla 1.4 y derivados. Aquí [inter.pdf] puede ver una transcripción rudimentaria en PDF, por si tiene dificultades. O bajarse todo [inter.zip], incluidas fuentes.

Introducción

Motivación

- **Acceso universal a programas, datos y servicios.**
- **Acceso a mercados globales.**
- **Creación de comunidades globales.**
- **Creación de mercados y comunidades locales.**
- **Cultura:**
  - **Idioma.**
  - **Creencias, usos y costumbres.**
  - **Regulaciones legales.**
  - **(Dis)capacidad física.**

Términos y siglas

i18n (internacionalización):	**Construir para la localización fácil.**
l10n (localización):	**Adaptar a culturas locales.**
Locale (localidad):	**Descripción de una cultura local.**
g11n (globalización):	**i18n + l10n**

¿Qué se localiza?

- **La interfaz de usuario:**
  - **Mensajes.**
  - **Iconos.**
- **Los objetos manipulados por el programa:**
  - **Documentos.**
  - **Datos.**

¿Como se localiza?

- **Estáticamente.**
- **Dinámicamente:**
  - **Tablas.**
  - **Bibliotecas dinámicas.**

Problemas fundamentales

- **Idioma:**
  - **Repertorio de caracteres.**
  - **Construcción de frases.**
  - **Ordenación.**
  - **Organización del espacio.**
  - **Procesamiento (búsqueda, normalización, pronunciación, ...)**
- **Cultural y legal:**
  - **Fechas, números, moneda, impuestos, etc.**
  - **Iconos significativos, no ofensivos.**

Interfaz de usuario

- **Mensajes de salida y error en catálogos.**
  - **¡No construirlos concatenando!**

```
"Mary" + "'s balance is USD" + "15"
"Mary" + "tiene un saldo de dólares" + "15"
```

  - **¡No basarse en ellos para guiones (scripts)!**
  - **Disminuye (un poco) la eficiencia.**
- **Mensajes de entrada, opciones.**
- **Tener en cuenta los límites espaciales.**
- **Los iconos deben ser significativos y no ofensivos: catálogos.**
- **Si hay E/S hablada: pronunciación.**

Interfaz de usuario

Clasificación de caracteres

- **Determinar legalidad, análisis léxico o sintáctico, búsquedas aproximadas.**
- **Predicados:**
  - **¿Es alfabético?**
  - **¿Es dígito?**
  - **¿Es espacio?**
  - **¿Es signo de puntuación?**
  - **¿Es mayúscula?**
  - **¿Es acentuada?**
  - **¿Es ligadura?**
  - **¿Es diacrítico?**
  - **¿Es ideograma?**

Trasliteración de caracteres

- **Convertir a mayúscula o a minúscula.**
  - **Sin sentido en árabe.**
  - **No tiene por qué ser trivial:**

```
#define toupper(c) ((c) - 'a' + 'A')
#define tolower(c) ((c) - 'A' + 'a')
```

  - **Utilizar tabla (puede ser muy grande).**
- **Quitar acento (u otro adorno).**

Ordenación de caracteres

- **Ayuda para la búsqueda: el código numérico puede valer:**
  - **En ASCII las mayúsculas salen antes que las minúsculas.**
- **Conviene usar el orden del diccionario:**
  - **Las letras acentuadas van juntas.**
  - **Los guiones no importan.**
  - **En alemán, la letra *ß* se ordena como *ss* (es una ligadura).**
  - **En castellano, el par *ch* se ordena(ba) entre la *c* y la *d*.**
- **Los ideogramas se ordenan por:**
  - **Representación fonética.**
  - **Número de *radicales* o de *trazos*.**

Ordenación de caracteres

Búsqueda de cadenas

- **Codificación única (normalizada).**
  - **Letras acentuadas: *á* o *a′*, pero no ambas.**
  - **Ligaduras: ligar siempre o nunca (*ß* o *ss*).**
- **Problemas con códigos multiocteto.**
  - **Codificación sin estado (no ISO 2022).**
  - **Fácil sincronización (eg: multiocteto UTF-8).**
- **Búsquedas especiales:**
  - **Independiente de mayúsculas o minúsculas.**
  - **Independiente de acentos.**

Patrones (expresiones regulares)

- **Expresar conjuntos de caracteres (POSIX 1003.2)**
  - **Minúscula no es `[a-z]`, sino `[[:lower:]]`.**
  - **Letra no es `[A-Za-z]`, sino `[[:alpha:]]`.**
  - **Incluir caracteres múltiples: `[a-[.ch.]]`.**
- **Los comodines (?, *) deben referirse a caracteres (no a octetos).**

Fechas y horas

- **Transmitir y ¿mantener? hora universal.**
- **Presentar hora local (incluyendo cambios estacionales).**
- **Mostrar fecha con el calendario local.**
  - **España: *lun 13 dic 2004***
  - **USA: *Mon Dec 13 2004***
  - **Árabe:**
    - **Gregoriano: *13* ### *2004* (al revés)**
    - **Hégira: *22* #### *1483* (al revés)**
  - **Japón:**
    - **Gregoriano: *2003#1#13#*.**
    - **Emperador: *##15#1#13#* (##########).**

Representación de números

- **EEUU: 5,434.25**
- **España: 5.434,25**
- **Francia: 5 434,25**
- **Paises árabes: dígitos árabes o indis.**
- **Representación de los negativos:**
  - **-100**
  - **100-**
  - **(100)**

Representación de monedas

- **EEUU: USD 1,000**
- **España: 158.433,5 pta**
- **Francia: 1F56**

Los caracteres

Repertorios

- **Hello!**
- **¡Hola!**
- **E#o#an#o #iu#a#de**
- **Grüß Gott**
- **#### ###**
- **### ####**
- **####**
- **##########!**
- **##########, ##########**
- **##**

De baudot a UCS

- **Baudot: 5 bits**
- **ASCII: 7 bits**
- **Variantes nacionales de ISO-8859: 8 bits**
- **Repertorios orientales: 16 bits.**
- **UCS / ISO 10646 / Unicode [http://www.unicode.org]**
  - **Unificación HAN # 16 bits (BMP)**
  - **Planos adicionales # 31 bits**

Unicode

- **Propiedades de los caracteres.**
- **Caracteres de combinación.**
- **Repertorio del Web: HTML 4.0, XML, ECMAScript, Java, ...**
- **Esquemas de codificación:**
  - **UCS-2 y UCS-4 para memoria.**
  - **UTF-7, *UTF-8* y UTF-16 (extensibles) para compatibilidad, transmisión y almacenamiento.**
  - **Repertorios tradicionales + entidades de carácter.**

Caracteres y juegos de caracteres

Gran *confusión* de términos:

Carácter:

Objeto gráfico abstracto usado en un lenguaje escrito:

- **A veces tiene relación con los *fonemas* (letras, sílabas).**
- **A veces representa una *idea*.**
- **Casi siempre se extiende con controles diversos: *retorno de carro, fin de línea, tabu-***

Caracteres y juegos de caracteres

lador, campana, sincronismos, porciones de protocolos, ...

Conjunto/juego/repertorio de caracteres:

Los usados en un idioma, conjunto de idiomas o aplicación.

Representación gráfica de los caracteres

Glifo:

Objetos gráficos concretos de la escritura:

- **A veces se corresponden uno a uno con los caracteres.**
- **A veces un carácter tiene varias formas.**
- **A veces los caracteres se ligan en un glifo.**

Tipo, fundición, font:

Juego de glifos del mismo estilo...

- **Formas: vertical, cursiva, inclinada, ..**
- **Pesos: normal, ne-**

Representación gráfica de los caracteres

grita, ...

- **Anchura: estrecho, normal, ancho.**
- **Tamaños: escalables o fijos.**

Codificación de los caracteres 1

Charset = Juego de caracteres codificado = Tabla de caracteres = CCS:	**Repertorio de caracteres numerados.**
Punto = posición = code position:	**Número de un carácter en un juego de caracteres codificado.**
Forma de codificación (CEF):	**Función de puntos a secuencias de *unidades de código*.**
Esquema de codificación (CES):	**Función reversible de secuencias de *unidades de código* a *secuencias de octetos*.**
Página de códigos:	**Correspondencia en-**

Codificación de los caracteres [1]

**tre esquemas de co-
dificación y secuen-
cias de glifos[2] .**

[1] Ver modelo de Unicode [http://www.unicode.org/unicode/reports/tr17/].
[2] No estoy seguro, hay que verificarlo.

Entrada de caracteres

- **A veces se corresponden uno a uno las teclas con los caracteres.**
- **A veces un carácter es combinación modificador y tecla.**
- **A veces un carácter es secuencia de tecla muerta y tecla.**
- **A veces se usan *métodos de entrada* más complejos:**
  - **Basados en punto.**
  - **Basados en menús (fonéticos).**

Esquemas de codificación sencillos

- **Cada carácter se representa en binario por su posición en el juego de caracteres codificado.**
- **Códigos sucesivos para caracteres sucesivos:**
  - **Ordenación de cadenas de caracteres.**
  - **Conversiones de cadenas de dígitos a números y viceversa.**
- **Necesidad de mayúsculas y minúsculas.**
- **Facilidad de conversión: mayúsculas / minúsculas.**
- **Ejemplos:**
  - **Baudot (5 bits), ASCII (7 bits), EBCDIC de IBM (7 bits)...**

ASCII y variantes de 7 bits

- **Permite representar minúsculas.**
- **Ordenación sencilla (códigos ascendentes).**
- **Conversiones y atributos sencillos (un bit).**
- **Caracteres de control.**
- **Sin diacríticos ni otros caracteres latinos # Variantes nacionales ISO 646.**

- **Incompatibles.**
- **Pierden signos de puntuación.**
- **Casi compatibles ASCII.**
- **Pierden facilidad de ordenar, convertir, clasificar...**

Esquemas de codificación sencillos de 8

- **7 bits bastan para los alfabetos latinos nacionales.**
- **Extensiones para representar símbolos gráficos: CP437, CP850 (Llamados ASCII por los usuarios de PC).**
- **A 10 de los 20 idiomas más hablados les bastan 8 bits.**
- **No bastan 8 bits para todos los alfabetos latinos.**
- **8 bits son un problema para canales de comunicación de 7 bits.**
  - **Ciertas conexiones asíncronas (7 bits + paridad).**
  - **Ciertos agentes de correo electrónico y otros protocolos de internet.**

ISO 8859

- **8 bits**
- **Compatibles con ASCII**
- **Preferidos en el correo electrónico MIME.**
- **10 variantes.**
- **Todos *contienen* ASCII.**
- **Latin-1 fué el código oficial de HTML (2.0).**

Variantes de ISO 8859

- **ISO 8859-1: Idiomas del oeste europeo (Latin-1)**
- **ISO 8859-2: Idiomas del este europeo (Latin-2)**
- **ISO 8859-3: Idiomas europeos de sudeste, esperanto (Latin-3)**
- **ISO 8859-4: Idiomas escandinavos/Balcanes (Latin-4)**
- **ISO 8859-5: Latin/Cirílico**
- **ISO 8859-6: Latin/Árabe**
- **ISO 8859-7: Latin/Griego**
- **ISO 8859-8: Latin/Hebreo**
- **ISO 8859-9: Modificación de Latin-1 para el Turco (Latin-5)**
- **ISO 8859-10: Idiomas Lapón/Nórdicos/Esquimal (Latin-6)**

Variantes de ISO 8859

- **ISO 8859-15: Idiomas del oeste europeo con € y ¢ (Latin-9).**

Idiomas más importantes

- ***Chino:*** **885.000.000 hablantes.**
- ***Inglés:*** **450.000.000 hablantes.**
- ***Hindi-Urdu:*** **333.000.000 hablantes.**
- ***Español:*** **266.000.000 hablantes.**
- ***Portugués:*** **175.000.000 hablantes.**
- ***Ruso:*** **153.000.000 hablantes.**
- ***Árabe:*** **150.000.000 hablantes.**
- ***Japonés:*** **126.000.000 hablantes.**
- ***Francés:*** **122.000.000 hablantes.**
- ***Alemán:*** **118.000.000 hablantes.**
- ***Bengalí:*** **110.000.000 hablantes.**
- ***Wu:*** **77.000.000 hablantes.**
- ***Javanés:*** **75.000.000 hablantes.**

Idiomas más importantes

- ***Coreano:*** **72.000.000 hablantes.**
- ***Italiano:*** **63.000.000 hablantes.**
- ***Marathi:*** **65.000.000 hablantes.**
- ***Telugu:*** **55.000.000 hablantes.**
- ***Tamil:*** **48.000.000 hablantes.**
- ***Cantonés:*** **47.000.000 hablantes.**
- ***Ucraniano:*** **46.000.000 hablantes.**

Nota

De BABEL/Alis [http://alis.isoc.org/langues/grandes.htm].

Extensiones del juego de caracteres

- **Secuencias de escape.**
- **Muy usados en terminales.**
- **ISO 2022:**
  - **Dos juegos de 16 caracteres de control: C0 y C1.**
  - **Cuatro juegos de 94 caracteres gráficos: G0, G1, G2, G3.**
  - **Operaciones para:**
    - **cambiar de juego activo (shift in, shift out)**
    - **cargar juego (escape ...)**
  - **Muy complicado al necesitar mantener estados.**
  - **Poco usado para manipular texto multilingüe.**

ISO10646 o UCS

- **Consenso de organismos de normalización.**
- **Contentar a todas las partes.**
- **Espacio de 32 bits.**
  - **256 grupos de 256 planos asignados a comités nacionales.**
  - **Primer plano: básico multilingüe, único asignado.**
- **Inmanejable.**
- **No define un orden de transmisión en octetos.**

Caracteres de ISO10646 o UCS

- **Introducidos: Latinos, Griego, Cirílico, Hebreo, Árabe, Armenio, Gregoriano, Japonés, Chino, Hiragana, Katakana, Coreano, Hanguliano, Devangari, Bengalí, Gurmuki, Gujarati, Oriya, Tamil, Telugu, Kannada, Malayam, Tai, Lao, Bopomofo, y algunos otros.**
- **Trabajando en: Tibetano, Kumer, Rúnico, Etíope, Jeroglíficos y varios idiomas indo-europeos.**
- **No alfabéticos: símbolos gráficos, tipográficos, matemáticos y científicos, como los proporcionados por TeX, PostScript, MS-DOS, Macintosh, Videotext, OCR, ...**

Unicode

- **Alianza de fabricantes.**
- **Convergencia con plano multilingüe de ISO10646 (inicialmente).**
- **Los 256 primeros puntos son Latin-1.**
- **Principio: unificación # mismo punto para caracteres parecidos.**
  - **Unificación HAN: Chino, Japonés, Coreano**
- **En evolución (actualmente versión 4.0, Mayo de 2002).**
- **FFFE y FEFF para detectar orden de transmisión.**
- **E000 a F8FF de libre uso (usuarios y fabricantes).**

Grados de realización de Unicode

Grado 1	**No combinación ni Hangul Jamo.**
Grado 2	**Algunos de combinación: Hebreo, Árabe, Devangari, Bengalí, Gurmukhi, Oriya, Tamil, Telugo, Kannada, Malayalam, Tai, y Lao.**
Grado 3	**Todos.**

Arquitectura de unicode

Nota

Verlo con gucharmap o en el consorcio Unicode [http://www.unicode.org/charts/] .

Esquemas de codificación multibyte

- **16 bits ocupan el doble, pero manejables.**
- **Muy incompatible con sistemas de 8 bits.**
- **Difícil internacionalizar programas hechos.**
- **Problemas de transmisión.**
- **Codificar caracteres más frecuentes con 8 bits y extender.**

Esquemas de codificación multibyte

- Problemas de
  - tamaño de campos.
  - cursores.
  - cortar y pegar.
  - edición de líneas.
  - casar patrones.
- Problemas del sistema de ficheros, etc...: segundo octeto "/".
- Problemas con fin de cadena (nulo).
- Problemas de sincronización.
- Solución: códigos extendidos que:
  - no contienen bytes que son códigos ASCII.
  - no contienen caracteres prefijos de otros.

Esquema de codificación UTF-8

- **Puntos 0 a 7F: un octeto y tal cual.**
- **Puntos mayores que 7F: como secuencias de 80-FD: elimina problemas de fin de cadena, sistema operativo, sincronización.**
- **Preserva orden de UCS.**
- **C0 a FD comienzan secuencia multiocteto: determinan longitud.**
- **El resto son siempre de 80 a BF.**
- **Representación (la más corta posible).**
- **FE y FF no se usan.**

Rangos en codificación UTF-8

- **Unicode**

```
0x00000000 - 0x0000007F: 0xxxxxxx
0x00000080 - 0x000007FF: 110xxxxx 10xxxxxx
0x00000800 - 0x0000FFFF: 1110xxxx 10xxxxxx 10x
```

- **Resto de UCS**

```
0x00010000 - 0x001FFFFF: 11110xxx 10xxxxxx 10x
0x00200000 - 0x03FFFFFF: 111110xx 10xxxxxx 10x
0x04000000 - 0x7FFFFFFF: 1111110x 10xxxxxx 10x
```

Ejemplos de codificación UTF-8

- **Copyright (©):**

```
0xa9 = 1010 1001
     -> 11000010 10101001 = 0xc2 0xa9
```

- **Desigual (#):**

```
0x2260 = 0010 0010 0110 0000
       -> 11100010 10001001 10100000 = 0xe2 0x8
```

UTF-16

- **Representa puntos de `0000` a `10FFFF`.**
- **Puntos menores que `10000` con 16 bits.**
- **Puntos mayores o iguales que `10000` con 32 bits.**
- **`D800` a `DFFF` para extender como UTF-8.**

UTF-7

- **Definido en RFC 1642, codifica UNICODE en 7 bits**
  - **Tal cual: `A--Z a--z 0--9 ' ( ) , - . / : ? +`**
  - **Opcionalmente: `! " # $ % & * ; < = > @ [ ] ^ _ ` { | } +`**
  - **Resto: +base64- (- no es necesario si no sigue de base64)**
  - **Signo +: +-**
  - **Ejemplo:**

Hi Mom #!	**hex**	**48, 69, 20, 4D, 6F, 4D, 20, 263A, 21**
	utf-7	**Hi Mom +Jjo-!**

Transferencia de información por canales

- **Puede usarse UTF-7 para Unicode.**
- **De hecho se usan métodos generales para datos de 8 bits:**
  - **Un volcado hexadecimal es poco compacto.**
    - **Expande 100%.**
  - **Histórico: uuencode para uucp.**
    - **Poco robusto cuando se sale de sistemas unix.**
    - **Expande 37%.**
  - **Modernamente base64 u otras (eg: base82 en PDF).**
    - **Expande 35%.**

Internacionalización en C y Unix

Caracteres estrechos y anchos

- **Caracteres estrechos, soportados por C (ISO 9899)**
  - **Octetos:**

```
char c = 'æ';
```

  - **Cadenas multiocteto:**

```
char mensaje[] = "ÁÉÍÓÚ ÀÈÌÒÙ";
```

  - **Caracteres anchos: tipos derivados y bibliotecas:**
    - **ISO C89**
    - **XPG2 / Unix 98**

Ejemplo: ISO C89

- **Caracteres anchos:**

```
wchar_t mensaje_ancho[];
```

- **Conversiones:**

```
mbstowcs (wchar_t *, const char *, size_t)
wcstombs (char *, const wchar_t *, size_t)
mbtowc (wchar_t *, const char *, size_t)
wctomb (char *, wchar_t)
...
```

- **Determinación de longitud:**

```
 int mblen (const char *, size_t)
```

- **`mbtowc` y `mblen` mantienen estado para juegos**

Ejemplo: ISO C89

con secuencias de conmutación (no reentrantes).

Localizaciones

- **Pasadas por variables de entorno o fijadas por programa.**
- **Las *localizaciones* tienen categorías con nombres**
  - **`LC_COLLATE`: Ordenación: `strcoll`.**
  - **`LC_CTYPE`: Clasificación, conversión.**
  - **`LC_MESSAGES`: Idioma de los mensajes (`catdoc` o `gettext`).**
  - **`LC_MONETARY`: Unidades monetarias (`localeconv`).**
  - **`LC_NUMERIC`: Otros números (`localeconv`, quizá `printf`).**
  - **`LC_TIME`: Impresión de fecha y hora (`strftime`).**
  - **`LC_ALL`: todos.**

Localizaciones

- **`LANG`: valor por omisión.**
- **`LANGUAGE`: lista de valores ordenada por preferencias.**

strftime

- ```
  size_t strftime(char *s, size_t max,
                  const char *format,
                  const struct tm *tm);
  ```

- **Mucha elección de formatos:**

  ```
  %c :     Representación preferida de día y hora
  %x :     Representación preferida de la fecha.
  %X :     Representación preferida de la hora.
  %a :     Abreviatura de día de la semana.
  %A :     Nombre de día de la semana.
  %b :     Abreviatura del mes.
  %B :     Nombre del mes.
  %p :     `am'  o  `pm'  o lo que sea...
  %d :     Número de día del mes.
  %H :     Hora (de 24).
  ```
  ```

strftime

localeconv

- ```
  struct lconv *localeconv(void);
  ```

- ```
  struct lconv {
    char *decimal_point;    /* Caracter usado co
    char *thousands_sep;    /* Separador de mile
    char *grouping;         /* Agrupamiento de d
    char *int_curr_symbol;  /* Basado en ISO 421
    char *currency_symbol;  /* Símbolo monetario
    char *mon_decimal_point;/* Caracter usado co
    char *mon_thousands_sep;/* Separador de mile
    char *mon_grouping;     /* Igual que el camp
    char *positive_sign;    /* Signo para valore
    char *negative_sign;    /* Signo para valore
    char int_frac_digits;   /* Dígitos fraccion
    char frac_digits;       /* Dígitos fraccion
    ...
  ```

localeconv

langinfo

- ```
  char * nl_langinfo (nl_item ITEM)
  ```
- ```
  strftime (s, len, "%X %D", tp);
  ```

  produce

  ```
  08:53:09 03/15/00
  ```
- **Lo correcto es**

  ```
  strftime (s, len, nl_langinfo (D_T_FMT), tp);
  ```

Ajuste de localizaciones

- C especifica que por omisión se arranaca en el `C`.
- Por lo menos `setlocale (LC_ALL, "");`
- Ajuste fino: `setlocale (CATEGORÍA, VALOR)`.
- Consulta: `setlocale (CATEGORÍA, NULL)`.
- Valores
  - `"C"` o `"POSIX"`: obligatorio y por omisión.
  - `""`: variable de entorno.
  - Dependientes del sistema:
    - `"spanish"`
    - Códigos internacionales de idiomas: `"es_ES"`, `"en_GB"`, `"en_US"`.
    - Idem + nombres de códigos de caracteres: `"es_ES.ISO-8859-1"`.
  - Como ficheros precompilados (por ejemplo en

Ajuste de localizaciones

`/usr/lib/locale`).

Códigos de idiomas

Idioma	ISO639	ISO639-2
Árabe	ar	ara
Aymara	ay	aym
Catalán	ca	cat
Inglés	en	eng
Esperan-to	eo	epo
Castella-no	es	esl/spa

Códigos de países ISO3166

es_ES	**Castellano de España**
es_MX	**Castellano de México**
en_US	**Inglés de USA**
en_GB	**Inglés británico**

Catálogos de mensajes: X/Open XPG4

Catálogos de mensajes numerados por módulos

```
#include <stdio.h>
#include <nl_types.h>

#define SET 1
#define MSG_HELLO 1

nl_catd catfd;

int main (int argc, char **argv) {
    setlocale (LC_ALL, "");
    catfd = catopen (basename (argv [0]), MCLoadA
    printf (catgets (catfd, SET, MSG_HELLO, "hell
    catclose (catfd);
    return 0;
}
```

Catálogos de mensajes: Uniforum/

Indexación por mensaje original

```
#include <locale.h>
#include <gettext.h>
#include <stdio.h>

main()
{
    setlocale (LC_ALL, "");
    textdomain("helloprogram");
    printf(gettext("Hello, world\n"));
}
```

`catgets` y `gettext`

`catgets` **dificultad para mantener los números de los mensajes, según se añaden o eliminan en los fuentes.**

`gettext`

- **El programador no se preocupa del catálogo.**
- **Colisión de nombres: difícil porque conviene usar frases:**

```
gettext("A spring in the spring");
```

```
ge
ge
ge
ge
```

- **Acceso más lento (índices o tablas *hash*).**

Traducción de cadenas de formato

- **No construir plurales ni géneros por programa**
- **A veces hace falta reordenar para traducir**
- **Ordenación en `printf`:**

```
"String `%s' has %d characters\n"
"%2$d Zeichen lang ist die Zeichenkette `%1$s'"
```

Herramientas

- **Editores.**
- **Transcodificadores.**
- **Compiladores de localizaciones y bibliotecas.**
- **Ayudas a la internacionalización.**
- **Ayudas a la localización.**
- **...**

Transcodificadores

- **Transformación:**
  - **Código intermedio con todos los caracteres: A # UCS # B (e.g: `iconv, tcs`)**
  - **Concatenación de transformaciones conocidas (no siempre reversibles):**

    A # M# B

    A # X # Y # B

    (e.g: `recode` de François Pinard).

Compiladores de localizaciones

- **La localización depende de los convenios locales y el juego de caracteres.**
- **Debe escribirse en algo muy transportable: Eg: Subconjunto de 83 caracteres *invariantes* de ISO646.**
  - **Utilizable directamente en todas las `ISO646` nacionales, ISO8859-x, UTF-7, UTF-8, IBM-PC, Mac...**
  - **Utilizable traducido en todas las EBCDIC...**
- **Requiere describir todos los caracteres con los invariantes.**
- **Requiere describir los locales en función de esos caracteres.**

Descripción de caracteres

Glifo	RFC1345	UCS	ISO10646
¼	14	00bc	VULGAR FRACTION ONE QUARTE
½	12	00bd	VULGAR FRACTION ONE HALF
¾	34	00be	VULGAR FRACTION THREE QUAR
¿	?I	00bf	INVERTED QUESTION MARK
À	A!	00c0	LATIN CAPITAL LETTER A WIT
Á	A'	00c1	LATIN CAPITAL LETTER A WIT
Â	A>	00c2	LATIN CAPITAL LETTER A WIT
Ã	A?	00c3	LATIN CAPITAL LETTER A WIT
Ä	A:	00c4	LATIN CAPITAL LETTER A WIT
Å	AA	00c5	LATIN CAPITAL LETTER A WIT
Æ	AE	00c6	LATIN CAPITAL LETTER AE
Ç	C,	00c7	LATIN CAPITAL LETTER C WIT
È	E!	00c8	LATIN CAPITAL LETTER E WIT

Descripción de locales

```
LC_TIME
abday "<d><o><m>"; "<l><u><n>"; "<m><a><r>";/
      "<m><i><e'>";"<j><u><e>"; "<v><i><e>"; "<s>

day "<d><o><m><i><n><g><o>"; "<l><u><n><e><s>";/
    "<m><a><r><t><e><s>"; "<m><i><e'><r><c><o><l>
    "<j><u><e><v><e><s>"; "<v><i><e><r><n><e><s>"
    "<s><a'><b><a><d><o>"

LC_CTYPE
digit <0>;<1>;<2>;<3>;<4>;<5>;<6>;<7>;<8>;<9>
blank <SP>;<HT>;<NS>
space <SP>;<LF>;<VT>;<FF>;<CR>;<HT>;<NS>
upper <A>;<B>;<C>;<D>;<E>;<F>;<G>;<H>;<I>;<J>;<K>
      <O>;<P>;<Q>;<R>;<S>;<T>;<U>;<V>;<W>;<X>;<Y>
      <A/>>;<A?>;<A:>;<AA>;<AE>;<C,>;<E!>;<E'>;<E
      <I/>>;<I:>;<D->;<N?>;<O!>;<O'>;<O/>;<O?>;<O
```

Descripción de locales

El compilador de localizaciones de POSIX

- `localedef -i en_CA -f ISO_8859-1 en_CA`
-

```
-rw-rw-r-- 1 joaquin  profes  10399 Jan  7 08:0
-rw-rw-r-- 1 joaquin  profes  10940 Jan  7 08:0
-rw-rw-r-- 1 joaquin  profes     93 Jan  7 08:0
-rw-rw-r-- 1 joaquin  profes     27 Jan  7 08:0
-rw-rw-r-- 1 joaquin  profes    480 Jan  7 08:0
-rw-rw-r-- 1 joaquin  profes     42 Jan  7 08:0
```

- **O todo junto (`/usr/lib/locale/locale-archive`).**
- **Los fuentes se pueden encontrar en ftp://dkuug.dk/i18n/WG15-collection**

Correo y conferencias

Internacionalización del correo y las confe-

- **Fuera de línea.**
- **Sin posiblilidad de negociación entre agentes de usuario.**
- **Pueden negociar agentes de transferencia (insuficiente).**
- **Generalmente sólo se tiene en cuenta el esquema de codificación.**
- **Podría tenerse en cuenta el idioma si hay que**
  - **Dar formato (partir palabras, etc).**
  - **Comprobar palabras, sintaxis, estilo, etc.**
  - **Analizar el texto.**
  - **Dictar.**
- ***SOLUCIÓN:***
  - **Nombrar esquema de codificación.**

Internacionalización del correo y las confe-

- **Restringir la variablilidad.**

Mime 1.0 y ESMTP

- **Correo tradicional basado en RFC 822 / SMTP:**
  - **ASCII (7 bits).**
  - **Tamaño limitado de líneas.**
  - **Una línea con un punto termina el mensaje.**
  - **Algunos agentes de transferencia transforman el mensaje:**
    - **`ASCII <-> EBCDIC`.**
    - **`>From`**

Requisitos de Mime 1.0 y ESMTP

- **Mime 1.0**
  - **Enviar mensajes de cualquier tipo (no sólo texto).**
  - **Enviar mensajes compuestos (de varios tipos).**
  - **Enviar mensajes a trozos.**
  - **Resistir las peores pasarelas, pero aprovechar las buenas.**
  - **Poder enviar varios tipos de texto.**
  - **Poder seleccionar el alfabeto.**
  - **Compatible con RFC-822**
- **ESMTP.**
  - **Datos de 8 bits (EHLO, 8BITMIME).**
  - **Transparencia (SIZE).**

Mensaje RFC 822

- **RFC 822:**

```
Received: (from joaquin@localhost) by colibri.d
Date: Tue, 7 Jan 1997 12:50:37 +0100
From: Joaquin Seoane <joaquin@dit.upm.es>
Message-Id: <199701071150.MAA02005@colibri.dit
To: joaquin@colibri.dit.upm.es
Subject: probando

Hola
```

Mensaje Mime 1.0 transferible por ESMTP

- **Nuevas cabeceras**

```
Received: (from joaquin@localhost) by colibri.d
Date: Tue, 7 Jan 1997 12:50:37 +0100
From: Joaquin Seoane <joaquin@dit.upm.es>
Message-Id: <199701071150.MAA02005@colibri.dit
Subject: =?ISO-8859-1?Q?A=F1o_nuevo=2C_vida_nu
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; charset=ISO-8859-1
Content-Transfer-Encoding: 8BIT

Feliz año 1997.
```

Mensaje Mime 1.0 transferible por SMTP

- **Con 7 bits, expandiendo sólo los códigos mayores de 128:**

```
Received: (from joaquin@localhost) by colibri.d
Date: Tue, 7 Jan 1997 12:50:37 +0100
From: Joaquin Seoane <joaquin@dit.upm.es>
Message-Id: <199701071150.MAA02005@colibri.dit
Subject: =?ISO-8859-1?Q?A=F1o_nuevo=2C_vida_nue
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; charset=ISO-8859-1
Content-Transfer-Encoding: QUOTED-PRINTABLE

Feliz a=F1o 1997.
```

Tipos MIME 1.0 simples

- **Tipo y subtipo de contenido: `Content-Type`**
  - **`text/plain, text/richtext, text/enriched, text/html`**
  - **`image/gif, image/jpeg`, …**
  - **`audio/basic`**
  - **`video/mpeg`**
  - **`application/octet-string, application/postscript, application/msword, application/pdf, application/pgp-signature`, …**

Tipos MIME 1.0 compuestos o referencias

- ***multipart/mixed, multipart/alternative, multipart/parallel, multipart/related*.**
- ***message/rfc822, message/partial, message/external-body*.**

Parámetros de tipos texto MIME 1.0

- **Parámetros del tipo `text/*`: juego de caracteres**
  - **`US-ASCII`.**
  - **`ISO-8859-x`.**
  - **En la práctica se usan otros, incluidos `UTF-7` y `UTF-8`.**
  - **`UTF-7` y `UTF-8`.**
  - **En la práctica se usan otros, incluido `windows-1252`.**

Codificación de transferencia MIME 1.0

- Permite pasar distintos medios
- Puede cambiarla el agente de transferencia
- Independiente del tipo del contenido
- Valores
  - 7BIT
  - 8BIT
  - BINARY
  - QUOTED-PRINTABLE
  - BASE64
- No se han definido compresiones (sí en HTTP)

Mime.types, mailcap

- **Suele asociarse un tipo Mime a un sufijo de nombre de fichero al enviarlo (tabla `/etc/mime.types`).**

application/postscript	**ps,eps**
image/jpeg	**jpeg, jpg**
text/richtext	**rtx**

- **Suele configurarse el agente de correo (o navegador) para lanzar visores externos (tabla `/etc/mailcap`).**

application/postscript	**gv '%s'**
image/jpeg	**display '%s'**

Mime.types, mailcap

text/richtext	**shownonascii iso-8859-1 -e richtext -p**

El web

Internacionalización del WEB

- **Sistema en línea.**
- **Es posible la negociación.**
- **Aspectos:**
  - **HTTP 1.1**
  - **HTML 4.0, XHTML y XML.**
  - **URI/URL**
  - **Hojas de estilo.**
  - **....**

HTTP

- **Protocolo de transferencia de hipertexto.**
- **En uso las versiones 0.9, 1.0, y 1.1.**
- **Diseñado para transferir cualquier objeto (html, imágenes, vídeo, ...)**
- **Sin estado.**
- **Pensado para utilización de proxis con y sin cache, pasarelas.**
- **Permite la consulta y la manipulación.**
  - **Método GET.**
  - **Método HEAD.**
  - **Método POST.**
  - **Método PUT.**
  - **Método DELETE.**
  - **Método TRACE.**

HTTP

- **Pensado parcialmente para negociar contenidos.**
- **Utiliza cabeceras similares a MIME.**

Ejemplo de interacción HTTP

```
GET / HTTP/1.0
HTTP/1.0200 Document follows
MIME-Version: 1.0
Server: CERN/3.0
Date: Tuesday, 07-Jan-97 13:43:18 GMT
Content-Type: text/html
Content-Length: 3487
Last-Modified: Tuesday, 24-Sep-96 09:26:31 GMT
<HTML> <HEAD>
  <title>Página de bienvenida del DIT-UPM</title>
</HEAD>
<BODY BACKGROUND="/figures/patterns/bwbackg.gif"
      TEXT="#000000" LINK="#0000FF" VLINK="#0000A
<table border=0>
<tr><td> <img src="/figures/logos/dit90allsh.gif"
```

Algunas cabeceras

- Pueden ir en peticiones y respuestas (entidades)
- En las respuestas
  - `Content-Type: text/html; char-set=ISO-8859-4`
  - `Content-Encoding: gzip`
  - `Content-Length: 2438`
  - `Content-Language: es`
  - `Date: Tuesday, 07-Jan-97 13:43:18 GMT`
  - `Last-Modified: Tuesday, 07-Jan-97 13:43:18 GMT`
  - `Expires: Tuesday, 07-Jan-97 13:43:18 GMT`
  - `Server: CERN/3.0 libwww/2.17`

Algunas cabeceras en peticiones

- **En las peticiones**

```
From: joaquin@dit.upm.es
Referer: http://www.dit.upm.es/sitios.html
If-Modified-Since: Tuesday, 07-Jan-97 13:43:18
Authorization: Basic QWxFGHvhfvGHHFgXXtTiI==
User-Agent: CERN-LineMode/2.15 libwww/2.17b3
Accept: text/plain; q=0.5, text/html, text/x-dv
Accept-charset: iso-8859-5, unicode-1-1;q=0.8
Accept-encoding: compress,gzip
Accept-language: da,en-gb;q=0.8,en;q=0.7
```

Negociación

Un único URI con varias alternativas: ¿cómo se negocia?

- **Dirigida por servidor.**
- **Dirigida por agente.**
- **Otros mecanismos: cookies.**

Negociación dirigida por servidor

- El agente dice sus preferencias y capacidades (si quiere).
- El servidor selecciona automáticamente.
- Ventajas:
  - Una vuelta.
- Desventajas:
  - Ineficiente por:
    - La longitud de las peticiones.
    - La escasez de recursos negociables.
    - Difunde información que puede ser confidencial.
  - El servidor no sabe todas las capacidades del cliente o el uso (ver, imprimir...).
  - Complica el servidor.

Negociación dirigida por servidor

- **Dificulta caches.**
- **Ejemplo: Debian [http://www.debian.org]**

Negociación dirigida por agente

- El agente no dice sus preferencias y capacidades.
- El servidor manda lista de alternativas.
- El agente selecciona manual o automáticamente.
- Las altenativas las puede mandar un proxy.
- No está normalizada todavía (¿HTTP-NG?).
- Implementada en Apache 1.3.4, EmWeb, al menos.

Negociación transparente

- **Combina ambas**
- **Permite a las caches realizar negociación dirigida por servidor**
- **Distribuye la carga de la negociación**

HTTP 1.1

- **RFC - 2068 (Enero 1997)**
- **Mejoras**
  - **Tratamiento correcto de la negociación dirigi-da por servidor en caches y proxis.**
  - **Normaliza extensiones propietarias de HTTP 1.0.**
  - **Varias transacciones por conexión.**
  - **Versiones.**
  - **Servidores virtuales.**

Conversaciones HTTP 1.1

- **Petición:**

```
GET / HTTP/1.1
Host: www.unicode.org
```

- **Algunas cabeceras nuevas:**
  - `HTTP-Version: HTTP/1.1`
  - `Host: www.dit.upm.es`
  - `Cache-Control: no-cache`
  - `Cache-Control: max-age`
  - `Content-Version: "2.1.2"`
  - `Vary: accept,accept-language`
  - `Vary: *`
  - `Content-Location: http://www.dit.upm.es/index.html.es`

Conversaciones HTTP 1.1

- **Reservados:**
  - **`Alternates:` para negociación en cliente.**
  - **`Negotiate:` para expresar que el cliente acepta negociación.**

Negociación dirigida por servidor

- **El servidor origen debe mandar campo `Vary:` (si *cacheable*).**
- **La cache puede decidir si responde con la copia o redirige la petición.**
- **La selección se hace por alternativas y calidades.**
- **No decir calidad, supone 1.**
- **El servidor puede ponderar calidades:**
  - **Mala traducción.**
  - **Mala imagen.**
- **Se pueden incluir comodines.**
- **Ejemplos:**
  - **`Accept-charset: iso-8859-5, unicode-1-1;q=0.8`**
  - **`Accept: text/plain; q=0.5, text/html,`**

Negociación dirigida por servidor

```
   application/postscript; q=0.8
```

- `Accept: audio/*; q=0.2, audio/basic`
- `Accept-language: da,en-gb;q=0.8,en;q=0.7`

Negociación dirigida por cliente

- El servidor (origen o proxy) ofrece alternativas.
- Puede combinarse con preferencias del cliente.
- La cache puede hacer negociación dirigida por servidor.
- Petición:

```
GET /paper HTTP/1.1
Host: x.org
User-Agent: WutxaWeb/2.4
Negotiate:
```

Ejemplo respuesta intermedia

```
HTTP/1.1 300 Multiple Choices
Date: Tue, 11 Jun 1996 20:02:21 GMT
Alternates:
  {"paper.html.en" 0.9 {type text/html} {language
  {"paper.html.fr" 0.7 {type text/html} {language
  {"paper.ps.en"   1.0 {type application/postscri
Vary: negotiate, accept, accept-language
Content-Type: text/html
Content-Length: 227
<h2>Multiple Choices:</h2>
 <ul>
   <li><a href=paper.html.en>HTML, English versio
   <li><a href=paper.html.fr>HTML, French version
   <li><a href=paper.ps.en>Postscript, English ve
 </ul>
```

Propuestas genéricas de cliente

- **Propuestas de *características* del cliente:**
  - **`tables paper=a4`**
  - **`frames colordepth=5`**
  - **`textonly pagewidth=200`**
- **El servidor puede generar *al vuelo* la respuesta:**
  - **Basado en `User-Agent:`**
  - **Basado en `Accept-*:`**
    - **Transliteración automática.**
    - **Traducción de características.**
    - **Traducción automática.**
  - **Imposible listar las alternativas.**
  - **Difícil cachear.**
  - **El servidor puede mandar la lista de variables**

Propuestas genéricas de cliente

que controla.

HTML

- Creado por Tim Berners-Lee (CERN, 1991)
- HTML 2.0 (RFC1866, 1995)
- HTML3.2 (1997)
  - Applets, tablas, formularios, flujo de texto sobre imágenes.
  - Elementos de presentación (FONT).
- HTML 4.0 (1997) y 4.01 (1999) e ISO (2000/2003)
  - Internacionalización:
    - Juego de caracteres del documento: ISO 10646.
    - Idioma, direccionalidad,...
    - Ruby
  - Accesibilidad
    - Énfasis en estructura (no presentación).

HTML

- Mejoras en tablas, formularios, etc.
- Soporte general de documentos compuestos
- Soporte de hojas de estilo.
- Soporte para impresión.

XHTML

- **Transición hacia XML.**
- **XHTML 1.0 (2000)**
  - **Estricto**
  - **De transición**
  - **Con *frames*.**
- **XHTML 1.1 (2001)**
  - **Modular**
- **XHTML 2.0 (borrador 2004)**
  - **Incompatible**

Entidades tipo carácter

- **Ahora todo ISO 10646**
  - **Representables como `&#0;` hasta `&#2147483645;`**
  - **Soporte de partición de líneas:**
    - **Blanco no partible: ` `**
    - **Indicación de donde partir: `` .**
  - **Soporte para enlace de cursivas**
    - **Separador sin espacio: `‌`**
    - **Juntador sin espacio: `‍`**
  - **Soporte para texto bidireccional (por defecto UNICODE)**
    - **Marca de izquierda a derecha: `‎`**
    - **Marca de derecha a izquierda: `‏`**
    - **Fijar direccionalidad...**

Entidades tipo carácter

- **Anidar direccionalidad...**

Atributos de idioma

- **Atributo LANG aplicable a casi cualquier elemento:**

```
<P LANG=es-ES>Esto es un
    <STRONG LANG="en">silly example</STRONG>
    de marcaje por idioma.
</P>
```

- **Atributo `xml:lang` en XML (y XHTML).**

```
<p>The title in Chinese is
    <span lang="zh-guoyu" xml:lang="zh-guoyu">
</p>
```

Atributos de direcionalidad

- **Atributo DIR (LTR o RTL) para fijar direccionalidad (a veces redundante con UNICODE)**

```
    ABxyCDzwEF # AByxCDwzEF


<SPAN DIR="LTR">AB
   <SPAN DIR="RTL">xy
      <SPAN DIR="LTR">CD</SPAN>zw
   </SPAN>EF
</SPAN>
  #    ABwzCDxyEF
```

Citas cortas, superíndices

- Citas cortas: `<Q>Esto es una cita</Q>`
  - "Esto es una cita"
  - `Esto es una cita'
  - «Esto es una cita»
- Superíndices/subíndices: `M<SUP>lle</SUP>` Dü-pont

Formularios

- **Atributo ACCEPT-CHARSET en INPUT y TEXTA-REA**
- **Usar método POST con tipo MIME `multipart/form-data`**

Hojas de estilo

- **HTML se ha degradado por el interés en el diseño.**
- **El estilo debe especificarse aparte**
  - **Contrarestar tendencia a poner estilo en HTML**
  - **Especificable por el usuario**
  - **Recomendable por el publicador**
- **Deben poderse especificar estilos alternativos, dependientes del medio (posiblemente parametrizados):**
  - **screen**
  - **tty**
  - **print**
  - **braille**
  - **aural**

CSS

- **CSS, CSS2.**
- **Anidables (usuario, enlaces, detalles).**
- **Un usuario puede elegir o deshabilitar los del autor.**
- **Permite ajustar tipos, colores tamaños, separaciones, márgenes, sonidos, fondos visuales y musicales, voces.**
- **Estilos, fondos, sujetos a negociación de idioma.**
- **Asociados a elementos (anidados o no) y clases**
- **Ejemplo:**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 T
<html>
  <meta http-equiv="Content-Type" content="tex
                                  charset=UTF-
```

CSS

```
  <head>
    <title>Título</title>
    <link rel=stylesheet type="text/css" media=
          href="http://www.dit.upm.es/~joaquin/
    <style type="text/css">
      h1 {color: blue}
      h2 {color: green}
    </style>
  </head>
  <body>
    <h1>La cabecera es azul</h1>
    <p style="color:red">Y el párrafo es rojo.<
    <p>Y este del color por omisión.</p>
  </body>
</html>
```

Ver [estilo.html]

Las hojas de estilo CSS2

```
body {color: blue; font-size: 26pt}
:lang(ar) {color: red}
:lang(zh-Hans) {color: green}


<p>It is polite to welcome people in their own la
<ul>
  <li xml:lang="zh-Hans" lang="zh-Hans">##</li>
  <li xml:lang="el"      lang="el">############</
  <li xml:lang="ar"      lang="ar">#### #####</li
  <li xml:lang="ru"      lang="ru">##### ########
</ul>
```

Ver [lang.html]

Los URL

- **IDN: Unicode # Normalización # Conversión a subconjunto de ASCII**
  - **http://bücher.ch # http://xn--bcher-kva.ch**
  - **http://#####.w3.mag.keio.ac.jp**
  - **RFC's 3490, 3491, 3492**
- **Parte local: Suconjunto imprimible de ASCII (RFC2936):**
  - **Permite enviar *octetos* arbitrarios en hexadecimal (e.g.: `%0D%0A%20`).**
    - **Caracteres de control.**
    - **Caracteres inseguros: `"<>{}|\^~[]`.**
    - **Caracteres reservados: `;/?:@=&%`.**
  - **Nada se dice del código de los caracteres (no se puede imprimir cómodamente)**

Los URL

- **RFC 2718: usar RFC2936 con UTF-8.**

Iniciativa WAI [http://www.w3.org/WAI]

- Iniciativa de accesiblildad en el WEB (Abril 1997)
- Recomendaciones para HTML (mejor HTML 4)
  - Formateo lógico
    - Anidar correctamente
    - Usar hojas de estilo
    - Utilizable sin interpretar el estilo
    - Facilita herramientas
  - Imágenes con texto alternativo o con explicaciones
  - Mapas de imagen en cliente o lista de alternativas
  - No poner texto en imágenes
  - No poner texto en columnas múltiples
  - Enlaces descriptivos y separados

Iniciativa WAI [http://www.w3.org/WAI]

- **Páginas alternativas**

Para saber más

Sitios web

- **Consorcio Unicode [http://www.unicode.org], The Unicode Standard V4.0 [http://www.unicode.org/versions/Unicode4.0.0/bookmarks.html] (publicado por Addison Wesley en 2003).**
- **Actividad i18n del W3C [http://www.w3c.org/international]**
- **i18n and Multilingual support in Internet mail [http://www.terena.nl/multiling/ml-mua/mldoc-review.html] (Yuri Demchenko, TERENA).**
- **Iniciativa Babel de ISOC/Alis [http://alis.isoc.org]**
- **Internacionalización de Java [http://java.sun.com/j2se/corejava/intl/index.jsp].**
- **Open directory of links to internationalization (i18n) resources and related material.**

Sitios web

[http://www.i18ngurus.com]

- **Internationalization (I18n), Localization (L10n), Standards, and Amusements [http://www.i18nguy.com].**

Libros

- **Yves Savourel, *XML Internationalization and Localization*, SAMS., 2001.**
- **Luong et al., *Internationalization: developing software for global markets*, John Wiley and Sons Inc., 1994.**
- **Galdo, *International user interfaces*, John Wiley and Sons Inc., 1996.**
- **XOPEN, *Internationalization Guide*, Ver. 2, Prentice Hall, 1995.**
- **Madell and Hewlett, *Developing and localizing international software*, Prentice Hall, 1994.**
- **Schmitt, *Designing international software*, Prentice Hall, 1996.**
- **Uren et al. *Software Internationalization and localization*, International Thomson Pub, 1993.**

Libros