

Cadenas de CARACTERES (strings)

Avda Brasil 2241

¿Qué tal?

ici1142

BBCL 34

AGUA sin GAS

#informaticaPUCV

JUAN perez

STRINGS

+ secuencia de cero o más
CARACTERES.

+ termina con un CARACTER especial
que marca el fin del string.

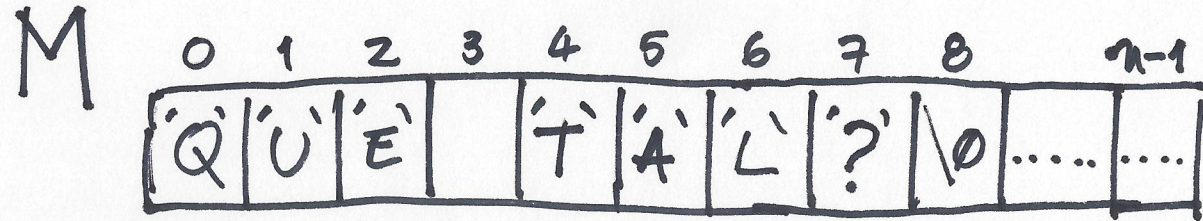
'\0'



+ para definir un string se usa
un arreglo de CARACTERES.

STRINGS

Qus tal?



fin string

char M[n] = "Qus tal?";

STRINGS

entrada

salida

scanf(...);
fgets(...);

fputs(...);
printf(...);

STRINGS

```
char *fgets (char *s, int n, FILE *stream);
```

←
arreglo donde se
almacenarán los
caracteres de la
cadena

↓
#max de
CARACTERES - 1,
agrega '\0' al final
de la cadena

↓
stdin

STRINGS y funciones

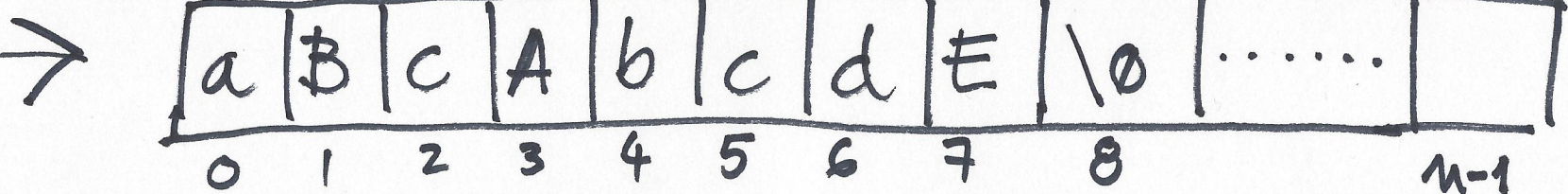
```
char s[n] = "aBcAbcdE";
```

⋮

```
mayuscula(s);
```

```
printf("%s\n", s); ⇒ "ABCABCDE"
```

S



STRINGS y Funciones

```
void mayuscula (char *s) {  
    for (i=0; s[i] != '\0'; i++)  
        s[i] = toupper(s[i]);  
}
```

#include <ctype.h>

toupper(...)	isupper(...)	isdigit(...)
tolower(...)	islower(...)	isspace(...)

STRING

(Ejercicio LAMBDA)

"VALDIVIA" < "VALDIVINOS"

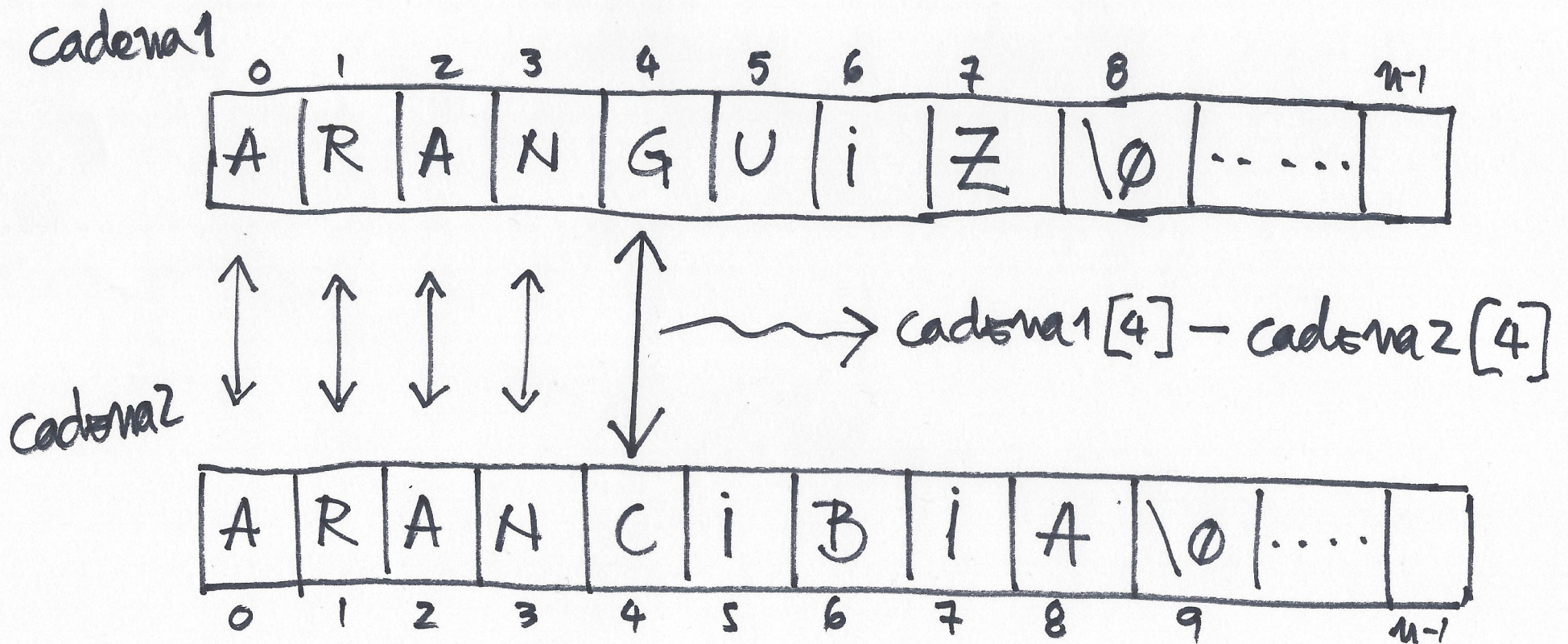
"HOLA" == "HOLA"

"PUERTO" > "PUERTA"

IMPORTANTE: 'A' ≠ 'a', en algunos casos
conviene transformar los strings
a comparar a mayúscula/minúscula.

STRINGS

(ejercicio LAMBDA)



STRINGS (EJERCICIO LAMBDA)

```
int mi_strcmp(.....) {
```

```
int i;
```

```
for (i=0; (cadena1[i]==cadena2[i]) && (cadena1[i]!='\0'); i++);
```

```
return cadena1[i] - cadena2[i];
```

```
}
```