Clase 01 - Introducción

IIC1001 - Algoritmos y Sistemas Computacionales

Cristian Ruz – cruz@uc.cl

Martes 10-Marzo-2026

Departamento de Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Contenidos

Presentación

Contacto

Temas

Para partir...

Contenidos

Presentación

Contacto

Temas

Para partir...

¡Bienvenidos y bienvenidas!

¡Bienvenidas y bienvenidos!





```
from Antofagasta import Cristian_Ruz
```

- Mi número de alumno es 97624632
- Ingeniero Civil de Computación, PUC (2005)
- Magister en Ciencias de la Ingeniería, PUC (2005)
- Doctorat en Informatique, Université Nice Sophia-Antipolis, Francia (2011)
- Profesor UC desde 2015, part-time desde 2013
- Profesor de IIC1001 por 4ta vez

¡Bienvenidos!









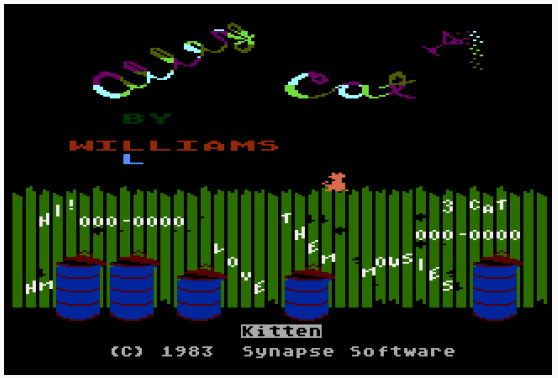
```
from Antofagasta import Cristian_Ruz
```

- Casado con Carla y papá de Camilo
- Primer computador: ZX Spectrum (1985), Z80 @ 3.5MHz, 16K RAM
- Atari 130XE, IBM PS/2, 80486, Pentium ...etc
- Lenguajes: C, C++, Java, Javascript, PHP, shell scripts, ...Python
- Cobreloa









Contenidos

Presentación

Contacto

Temas

Para partir...

Contacto

🐐

vicente.cabra@uc.cl

Vicente Cabra

Ayudante jefe



🐸

alejandro.tapia@uc.cl

Alejandro Tapia

Ayudante jefe



🥇

gabriel.sandoval@uc.cl

Gabriel Sandoval

Ayudante jefe



Contenidos

Presentación

Contacto

Temas

Para partir...

Temas del curso

Sistemas computacionales

- Representación datos, números y compresión
- Funcionamiento hardware, procesadores y memoria.
- Funcionamiento de sistemas operativos: ejemplo scheduling
- Funcionamiento de Internet
- Herramientas computacionales: github + latex

Algoritmos

- Algoritmos y resolución de problemas
- Eficiencia algorítmica
- Estructuras secuenciales y ordenamiento
- Grafos y árboles

Contenidos

Presentación

Contacto

Temas

Para partir...

¿Qué hace este algoritmo?

Algorithm 1.1: A simple Stock Span algorithm.

```
SimpleStockSpan(quotes) → spans
    Input: quotes, an array with n stock price quotes
    Output: spans, an array with n stock price spans

1   spans ← CreateArray(n)
2   for i ← 0 to n do
3       k ← 1
4       span_end ← FALSE
5       while i − k ≥ 0 and not span_end do
6           if quotes[i − k] ≤ quotes[i] then
7               k ← k + 1
8           else
9               span_end ← TRUE
10      spans[i] ← k
11  return spans
```

Un programa

Un programa ... en C

```c
#include <stdio.h>

int main() {
    printf("Hello, world\n");
    return 0;
}
```

Un programa

Un programa … en bits

```
00100011 01101001 01101110 01100011 01101100 01110101 01100100 01100101
00100000 00111100 01110011 01110100 01100100 01101001 01101111 00101110
01101000 00111110 00001010 00001010 01101001 01101110 01110100 00100000
01101101 01100001 01101001 01101110 00101000 00101001 00100000 01111011
00001010 00100000 00100000 00100000 00100000 01110000 01110010 01101001
01101110 01110100 01100110 00101000 00100010 01001000 01100101 01101100
01101100 01101111 00101100 00100000 01110111 01101111 01110010 01101100
01100100 01011100 01101110 00100010 00101001 00111011 00001010 00100000
00100000 00100000 00100000 01110010 01100101 01110100 01110101 01110010
01101110 00100000 00110000 00111011 00001010 01111101
```

Un programa

Un programa … en caracteres

```
 #   i   n   c   l   u   d   e  <sp> <   s   t   d   i   o   .
 35 105 110  99 108 117 100 101  32  60 115 116 100 105 111 46

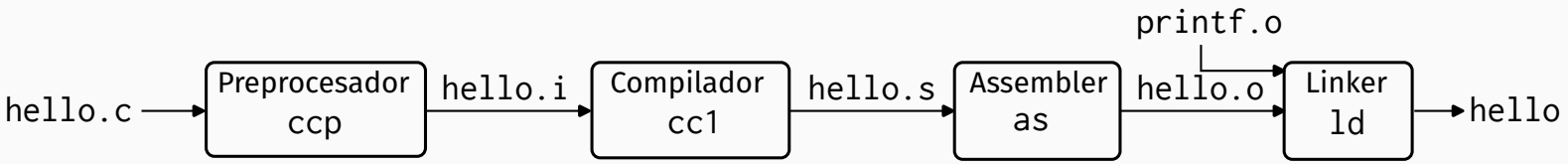
 h   >  \n  \n   i   n   t  <sp> m   a   i   n   (   )  <sp> {
104  62  10  10 105 110 116  32 109  97 105 110  40  41  32 123

\n  <sp><sp><sp><sp> p   r   i   n   t   f   (   "   H   e   l
 10  32  32  32  32 112 114 105 110 116 102  40  34  72 101 108

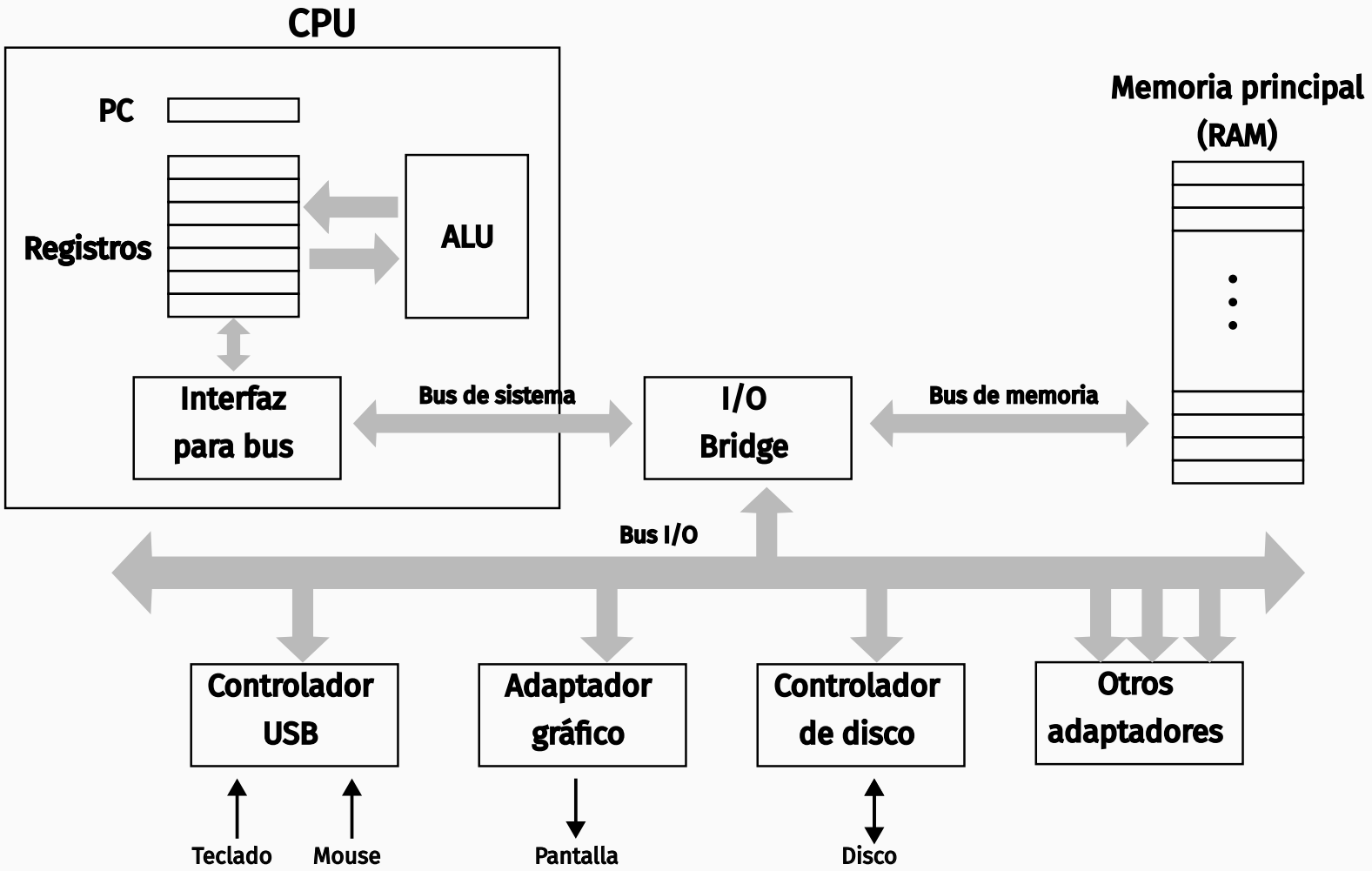
 l   o   ,  <sp> w   o   r   l   d   \   n   "   )   ;  \n  <sp>
108 111  44  32 119 111 114 108 100  92 110  34  41  59  10  32

<sp><sp><sp> r   e   t   u   r   n  <sp> 0   ;  \n   }  \n
 32  32  32 114 101 116 117 114 110  32  48  59  10 125 10
```

El proceso de compilación



Para llegar a ejecutar en ...



¡Disfruten el curso!