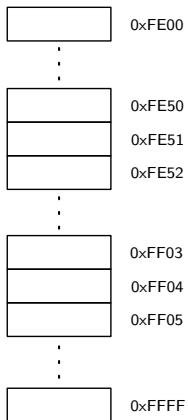
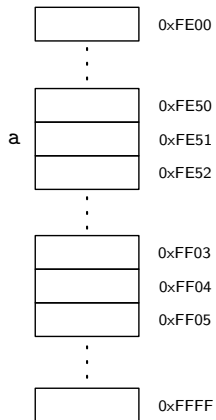


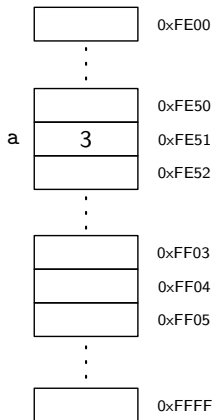
Informática I
Punteros - 1era Parte

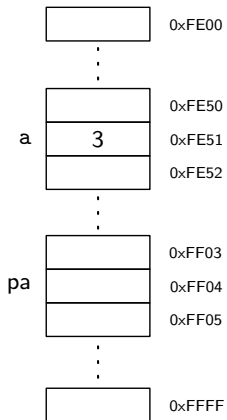
Claudio J. Paz

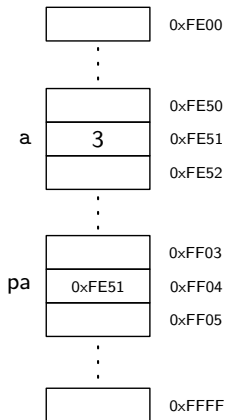
11 de septiembre de 2018



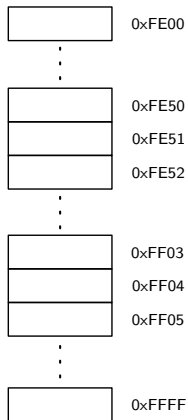




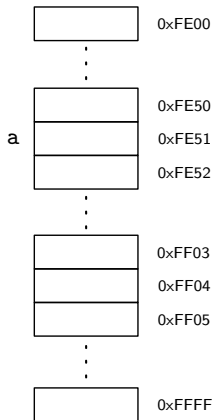




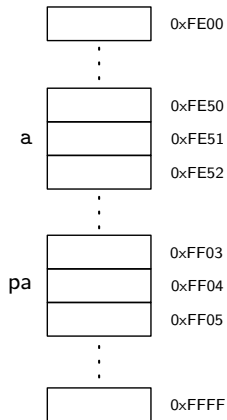
```
#include<stdio.h>
int main(void){
```



```
#include<stdio.h>
int main(void){
    int a;
```

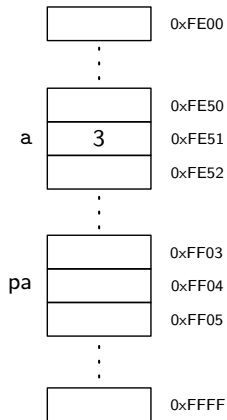



```
#include<stdio.h>
int main(void){
    int a;
    int *pa;
```



```
#include<stdio.h>
int main(void){
    int a;
    int *pa;

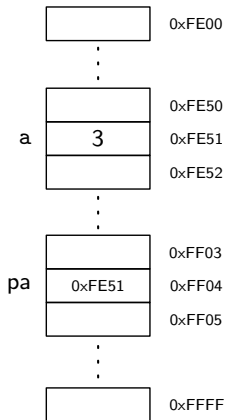
    a = 3;
```



```
#include<stdio.h>
int main(void){
    int a;
    int *pa;

    a = 3;

    pa = &a;
```



```

#include<stdio.h>
int main(void){
    int a;
    int *pa;

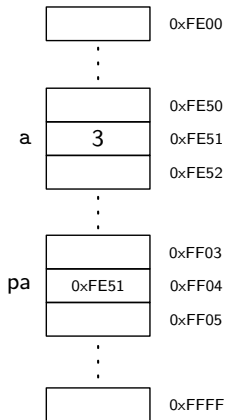
    a = 3;

    pa = &a;

    printf("La dirección de a es %p\n", &a);
    printf("El valor de pa es %p\n", pa);

    return 0;
}

```



```

#include<stdio.h>
int main(void){
    int a;
    int *pa;

    a = 3;

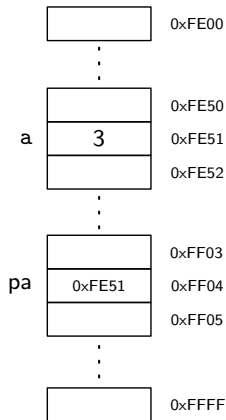
    pa = &a;

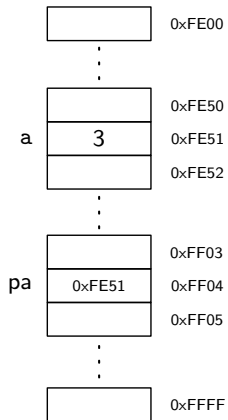
    printf("La dirección de a es %p\n", &a);
    printf("El valor de pa es %p\n", pa);

    return 0;
}

```

La dirección de a es 0xfe51
 El valor de pa es 0xfe51





```

#include<stdio.h>
int main(void){
    int a;
    int *pa;

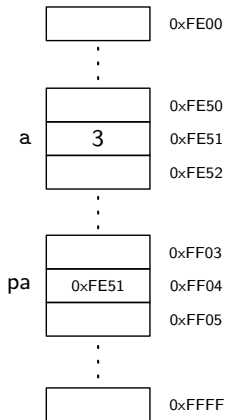
    a = 3;

    pa = &a;

    printf("          El valor de a es %d\n", a);
    printf("El valor que apunta pa es %d\n", *pa);

    return 0;
}

```



```

#include<stdio.h>
int main(void){
    int a;
    int *pa;

    a = 3;

    pa = &a;

    printf("          El valor de a es %d\n", a);
    printf("El valor que apunta pa es %d\n", *pa);

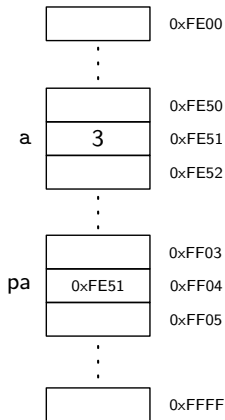
    return 0;
}

```

```

          El valor de a es 3
El valor que apunta pa es 3

```




```
int a,b;  
int *pa;
```

```
a = 3;
```

```
pa = &a;  
b = *pa;
```

```
int a,b;  
int *pa;  
  
a = 3;  
  
pa = &a;  
b = *pa;
```

- El & se llama Operador de Dirección u Operador de Referencia

```
int a, b;  
int *pa;  
  
a = 3;  
  
pa = &a;  
b = *pa;
```

- El & se llama Operador de Dirección u Operador de Referencia
- El * se llama Operador de Indirección u Operador de Desreferencia

Llamadas a funciones por valor

Llamadas a funciones por valor

```
#include <stdio.h>

int cuadrado(int x){
    x *= x;
    return x;
}

int main(void){
    int a;

    a = 3;

    printf("El valor de a es %d\n", a);
    printf("El cuadrado de a ");
    printf("es %d\n", cuadrado(a));
    printf("El valor de a es %d\n", a);

    return 0;
}
```

Llamadas a funciones por valor

```
#include <stdio.h>

int cuadrado(int x){
    x *= x;
    return x;
}

int main(void){
    int a;

    a = 3;

    printf("El valor de a es %d\n", a);
    printf("El cuadrado de a ");
    printf("es %d\n", cuadrado(a));
    printf("El valor de a es %d\n", a);

    return 0;
}
```

```
El valor de a es 3
El cuadrado de a es 9
El valor de a es 3
```

Llamadas a funciones por referencia

Llamadas a funciones por referencia

```
#include <stdio.h>

int cuadrado(int *x){
    *x *= *x;
    return *x;
}

int main(void){
    int a;

    a = 3;

    printf("El valor de a es %d\n", a);
    printf("El cuadrado de a ");
    printf("es %d\n", cuadrado(&a));
    printf("El valor de a es %d\n", a);

    return 0;
}
```


Llamadas a funciones por referencia

```
#include <stdio.h>

int cuadrado(int *x){
    *x *= *x;
    return *x;
}

int main(void){
    int a;

    a = 3;

    printf("El valor de a es %d\n", a);
    printf("El cuadrado de a ");
    printf("es %d\n", cuadrado(&a));
    printf("El valor de a es %d\n", a);

    return 0;
}
```

```
El valor de a es 3
El cuadrado de a es 9
El valor de a es 9
```

Llamadas a funciones por referencia

Llamadas a funciones por referencia

```
#include <stdio.h>

void cuadrado(int *x){
    *x *= *x;
}

int main(void){
    int a;

    a = 3;

    printf("El valor de a es %d\n", a);
    printf("El cuadrado de a ");
    cuadrado(&a);
    printf("es %d\n", a);

    return 0;
}
```

Llamadas a funciones por referencia

```
#include <stdio.h>

void cuadrado(int *x){
    *x *= *x;
}

int main(void){
    int a;

    a = 3;

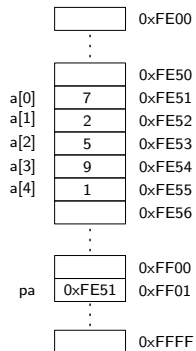
    printf("El valor de a es %d\n", a);
    printf("El cuadrado de a ");
    cuadrado(&a);
    printf("es %d\n", a);

    return 0;
}
```

```
El valor de a es 3
El cuadrado de a es 9
```

Aritmética de punteros

Aritmética de punteros



Aritmética de punteros

```
#include <stdio.h>

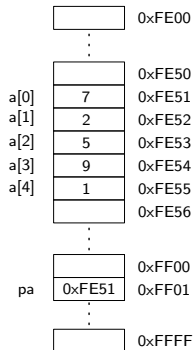
#define N 5

int main(void){
    char a[N] = {7,2,5,9,1};
    char *pa;

    pa = &a[0];

    printf("%d\n", *pa);

    return 0;
}
```



Aritmética de punteros

```
#include <stdio.h>

#define N 5

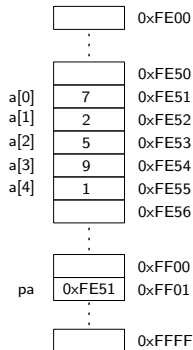
int main(void){
    char a[N] = {7,2,5,9,1};
    char *pa;

    pa = &a[0];

    printf("%d\n", *pa);

    return 0;
}
```

7



Aritmética de punteros

```
#include <stdio.h>

#define N 5

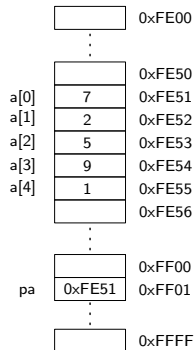
int main(void){
    char a[N] = {7,2,5,9,1};
    char *pa;

    pa = &a[0];

    pa += 1;

    printf("%d\n", *pa);

    return 0;
}
```



Aritmética de punteros

```
#include <stdio.h>

#define N 5

int main(void){
    char a[N] = {7,2,5,9,1};
    char *pa;

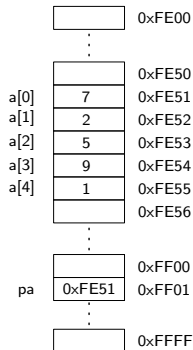
    pa = &a[0];

    pa += 1;

    printf("%d\n", *pa);

    return 0;
}
```

2



Aritmética de punteros

```
#include <stdio.h>

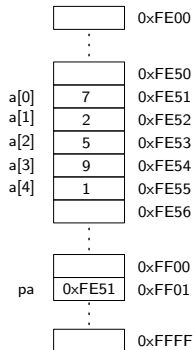
#define N 5

int main(void){
    char a[N] = {7,2,5,9,1};
    char *pa;

    pa = &a[0];

    printf("%d\n", *(pa+2));

    return 0;
}
```



Aritmética de punteros

```
#include <stdio.h>

#define N 5

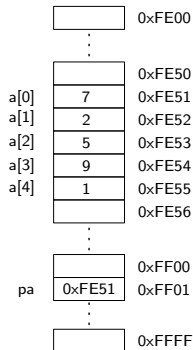
int main(void){
    char a[N] = {7,2,5,9,1};
    char *pa;

    pa = &a[0];

    printf("%d\n", *(pa+2));

    return 0;
}
```

5



Aritmética de punteros

```
#include <stdio.h>

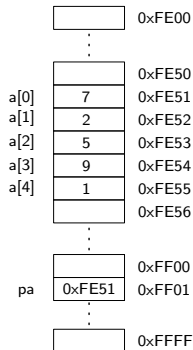
#define N 5

int main(void){
    char a[N] = {7,2,5,9,1};
    char *pa;

    pa = &a[0];

    printf("%d\n", *pa+2);

    return 0;
}
```



Aritmética de punteros

```
#include <stdio.h>

#define N 5

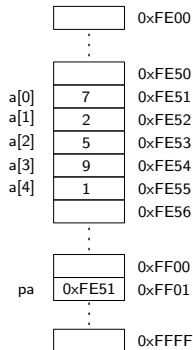
int main(void){
    char a[N] = {7,2,5,9,1};
    char *pa;

    pa = &a[0];

    printf("%d\n", *pa+2);

    return 0;
}
```

9



Aritmética de punteros

```
#include <stdio.h>

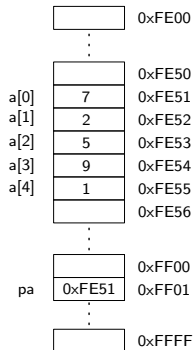
#define N 5

int main(void){
    char a[N] = {7,2,5,9,1};
    char *pa;
    int i;

    pa = &a[0];

    for ( i = 0; i < N ; i++ )
        printf("%d\n", *(pa+i));

    return 0;
}
```



Aritmética de punteros

```
#include <stdio.h>

#define N 5

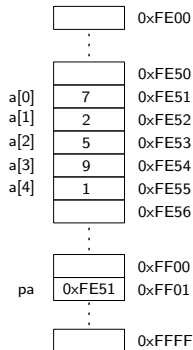
int main(void){
    char a[N] = {7,2,5,9,1};
    char *pa;
    int i;

    pa = &a[0];

    for ( i = 0; i < N ; i++ )
        printf("%d\n", *(pa+i));

    return 0;
}
```

```
7
2
5
9
1
```



Aritmética de punteros

```
#include <stdio.h>

#define N 15

int main(void){
    int i;
    char a[N] = {0};

    for ( i = 0; i < N ; i++ )
        printf("La dirección de a[%2d] es %p\n", i, &a[i]);

    return 0;
}
```

Aritmética de punteros

```
#include <stdio.h>

#define N 15

int main(void){
    int i;
    char a[N] = {0};

    for ( i = 0; i < N ; i++ )
        printf("La dirección de a[%2d] es %p\n", i, &a[i]);

    return 0;
}
```

```
La dirección de a[0] es 0x7fff4e408050
La dirección de a[1] es 0x7fff4e408051
La dirección de a[2] es 0x7fff4e408052
La dirección de a[3] es 0x7fff4e408053
La dirección de a[4] es 0x7fff4e408054
La dirección de a[5] es 0x7fff4e408055
La dirección de a[6] es 0x7fff4e408056
La dirección de a[7] es 0x7fff4e408057
La dirección de a[8] es 0x7fff4e408058
La dirección de a[9] es 0x7fff4e408059
La dirección de a[10] es 0x7fff4e40805a
La dirección de a[11] es 0x7fff4e40805b
La dirección de a[12] es 0x7fff4e40805c
La dirección de a[13] es 0x7fff4e40805d
La dirección de a[14] es 0x7fff4e40805e
```

Aritmética de punteros

```
#include <stdio.h>

#define N 15

int main(void){
    int i;
    int a[N] = {0};

    for ( i = 0; i < N ; i++ )
        printf("La dirección de a[%2d] es %p\n", i, &a[i]);

    return 0;
}
```

```
La dirección de a[0] es 0x7ffdbbf04090
La dirección de a[1] es 0x7ffdbbf04094
La dirección de a[2] es 0x7ffdbbf04098
La dirección de a[3] es 0x7ffdbbf0409c
La dirección de a[4] es 0x7ffdbbf040a0
La dirección de a[5] es 0x7ffdbbf040a4
La dirección de a[6] es 0x7ffdbbf040a8
La dirección de a[7] es 0x7ffdbbf040ac
La dirección de a[8] es 0x7ffdbbf040b0
La dirección de a[9] es 0x7ffdbbf040b4
La dirección de a[10] es 0x7ffdbbf040b8
La dirección de a[11] es 0x7ffdbbf040bc
La dirección de a[12] es 0x7ffdbbf040c0
La dirección de a[13] es 0x7ffdbbf040c4
La dirección de a[14] es 0x7ffdbbf040c8
```

Aritmética de punteros

```
#include <stdio.h>

#define N 5

int main(void)
{
    int i;
    char a[N] = {7,2,5,9,1};
    char *pa;

    pa = &a[0];

    for( i = 0; i < N ; i++)
        printf("%p\n", pa+i);

    return 0;
}
```

Aritmética de punteros

```
#include <stdio.h>

#define N 5

int main(void)
{
    int i;
    char a[N] = {7,2,5,9,1};
    char *pa;

    pa = &a[0];

    for( i = 0; i < N ; i++)
        printf("%p\n", pa+i);

    return 0;
}
```

```
0x7ffe3704a000
0x7ffe3704a001
0x7ffe3704a002
0x7ffe3704a003
0x7ffe3704a004
```

Aritmética de punteros

```
#include <stdio.h>

#define N 5

int main(void)
{
    int i;
    int a[N] = {7,2,5,9,1};
    int *pa;

    pa = &a[0];

    for( i = 0; i < N ; i++)
        printf("%p\n", pa+i);

    return 0;
}
```

Aritmética de punteros

```
#include <stdio.h>

#define N 5

int main(void)
{
    int i;
    int a[N] = {7,2,5,9,1};
    int *pa;

    pa = &a[0];

    for( i = 0; i < N ; i++)
        printf("%p\n", pa+i);

    return 0;
}
```

```
0x7ffc6ea69370
0x7ffc6ea69374
0x7ffc6ea69378
0x7ffc6ea6937c
0x7ffc6ea69380
```

Aritmética de punteros

```
#include <stdio.h>

#define N 5

int main(void)
{
    int i;
    long long int a[N] = {7,2,5,9,1};
    long long int *pa;

    pa = &a[0];

    for( i = 0; i < N ; i++)
        printf("%p\n", pa+i);

    return 0;
}
```


Aritmética de punteros

```
#include <stdio.h>

#define N 5

int main(void)
{
    int i;
    long long int a[N] = {7,2,5,9,1};
    long long int *pa;

    pa = &a[0];

    for( i = 0; i < N ; i++)
        printf("%p\n", pa+i);

    return 0;
}
```

```
0x7ffdd78a1c90
0x7ffdd78a1c98
0x7ffdd78a1ca0
0x7ffdd78a1ca8
0x7ffdd78a1cb0
```

Equivalencia entre punteros y arreglos

Equivalencia entre punteros y arreglos

```
#include <stdio.h>

#define N 5

int main(void){
    int i = 0;
    int a[N] = {7,2,5,9,1};
    int *pa;

    pa = &a[0];

    printf("%d\n", *pa);
}
```

Equivalencia entre punteros y arreglos

```
#include <stdio.h>

#define N 5

int main(void){
    int i = 0;
    int a[N] = {7,2,5,9,1};
    int *pa;

    pa = a;

    printf("%d\n", *pa);
}
```

Equivalencia entre punteros y arreglos

```
#include <stdio.h>

#define N 5

int main(void){
    int i = 0;
    int a[N] = {7,2,5,9,1};
    int *pa;

    pa = a;

    printf("%d\n", *(pa+i));
}
```

Equivalencia entre punteros y arreglos

```
#include <stdio.h>

#define N 5

int main(void){
    int i = 0;
    int a[N] = {7,2,5,9,1};
    int *pa;

    pa = a;

    printf("%d\n", pa[0]);
}
```

Consultas

claudiojpaz@gmail.com

Horario de Consulta: Martes 18:00-19:00hs
Of.5 Ed.Salcedo