

IMPLEMENTACION ESTATICA

```
typedef int ElementoP;  
typedef struct {  
    ElementoP datos[50];  
    int tope; } TPila;  
  
void poneP(TPila *P, ElementoP x) {  
    if ( ((*P).tope)!=49)  
        (*P).datos[++((*P).tope)] = x;  
}  
  
void sacaP(TPila *P, ElementoP* x) {  
    if (((*P).tope) != -1)  
        (*x) = (*P).datos[((*P).tope)--];  
}  
  
ElementoP consultaP(TPila P) {  
    if ((P.tope) != -1)  
        return P.datos[P.tope];  
}  
  
int VaciaP(TPila P) {  
    return (P.tope == -1);  
}  
  
void IniciaP (TPila *P) {  
    (*P).tope=-1;  
}
```

IMPLEMENTACION DINAMICA

```
typedef int ElementoP;  
typedef struct {  
    ElementoP dato;  
    struct nodop *sig; } nodop;  
typedef nodop *TPila;  
void poneP(TPila *P, ElementoP x) {  
    TPila N;  
    N = (TPila)malloc(sizeof(nodop));  
    N->dato = x;  
    N->sig = *P;  
    *P=N;  
}  
void sacaP(TPila *P, ElementoP * x) {  
    TPila N;  
    if (*P) { // if (*P != NULL)  
        N = *P;  
        *x = (*P)->dato;  
        *P = (*P)->sig;  
        free(N);  
    }  
}  
ElementoP consultaP(TPila P) {  
    if (P) // if (P != NULL)  
        return P->dato;  
}  
int VaciaP(TPila P) {  
    return (P == NULL);  
}  
void IniciaP (TPila *P) {  
    *P =NULL;  
}
```