

```

int lista_tamano(Lista *pl) {
    Nodo *pn;
    int tam;

    if ( pl == NULL ) {
        return -1;
    }

    if ( pl->first == NULL ) {
        return 0;
    }

    pn = pl->first;
    tam = 1;
    while ( pn->next != NULL ) {
        pn = pn->next;
        tam++;
    }

    return tam;
}

status lista_insertarEnPosicion(Lista *pl, Elemento *pe, int pos) {
    Nodo *pn, *pnodo;
    int i;

    if ( pl == NULL || pe == NULL || pos < 0 ) {
        return ERROR;
    }

    // Creamos el nodo a insertar en la lista
    pnodo = nodo_crear();
    if ( pnodo == NULL ) {
        return ERROR;
    }
    pnodo->info = elemento_copiar(pe);
    if ( pnodo->info == NULL ) {
        nodo_liberar(pnodo);
        return ERROR;
    }

    // Caso 1: lista vacía
    if ( pl->first == NULL ) {
        if ( pos == 0 ) {
            pl->first = pnodo;
            return OK;
        }
        else {
            nodo_liberar(pnodo);
            return ERROR;
        }
    }

    // Caso 2: lista no vacía, pos == 0
    if ( pos == 0 ) {
        pnodo->next = pl->first;
        pl->first = pnodo;
        return OK;
    }

    // Caso 3: lista no vacía, pos != 0
    pn = pl->first;
    i = 0;
    while ( pn->next != NULL ) {
        if ( i == pos ) {
            pnodo->next = pn->next;
            pn->next = pnodo;
            return OK;
        }
        else {
            i++;
            pn = pn->next;
        }
    }

    nodo_liberar(pnodo);
    return ERROR;
}

```

```

status lista_insertarEnOrden(Lista *pl, Elemento *pe) {
    Nodo *pn1, *pn2, *pnodo;
    int cmp;

    if ( pl == NULL || pe == NULL ) {
        return ERROR;
    }

    // Creamos el nodo a insertar en la lista
    pnodo = nodo_crear();
    if ( pnodo == NULL ) {
        return ERROR;
    }
    pnodo->info = elemento_copiar(pe);
    if ( pnodo->info == NULL ) {
        nodo_liberar(pnodo);
        return ERROR;
    }

    // Caso 1: lista vacía
    if ( pl->first == NULL ) {
        pl->first = pnodo;
        return OK;
    }

    // Caso 2: lista de 1 nodo
    if ( pl->first->next == NULL ) {
        if( elemento_comparar(pe, pl->first->info) {
            pnodo->next = pl->first;
            pl->first = pnodo;
        }
        else {
            pl->first->next = pnodo;
        }
        return OK;
    }

    // Caso 3: lista de 2 o más nodos
    pn1 = pl->first;
    if ( elemento_comparar(pe, pn1->info) < 0 ) {
        pnodo->next = pl->first;
        pl->first = pnodo;
        return OK;
    }
    pn2 = pl->first->next;
    while ( pn2 != null ) {
        if ( elemento_comparar(pe, pn2->info) < 0 ) {
            pnodo->next = pn2;
            pn1->next = pnodo;
            return OK;
        }
        pn1 = pn1->next;
        pn2 = pn2->next;
    }
    pn1->next = pnodo;

    return OK;
}

```

```

Lista *lista_concatenar(Lista *p11, Lista *p12) {
    Lista *p13;
    Nodo *pn, *pn3, *pnodo;

    if ( p11 == NULL || p12 == NULL ) {
        return NULL;
    }

    p13 = lista_crear();
    if ( p13 == NULL ) {
        return NULL;
    }

    pn3 = NULL;

    // Se copian los elementos de p11 a p13
    if ( p11->first != NULL ) {
        pn = p11->first;    // pn recorrerá p11
        pnodo = nodo_crear();
        pnodo->info = elemento_copiar(pn->info);
        if ( pnodo->info == NULL ) {
            nodo_liberar(pnodo);
            lista_liberar(p13);
            return NULL;
        }

        p13->first = pnodo;

        pn3 = pnodo;    // pn3 ecorrerá p13
        while ( pn->next != NULL ) {
            pn = pn->next;
            pnodo = nodo_crear();
            pnodo->info = elemento_copiar(pn->info);
            if ( pnodo->info == NULL ) {
                nodo_liberar(pnodo);
                lista_liberar(p13);
                return NULL;
            }
            pn3->next = pnodo;
            pn3 = pnodo;
        }
    }

    // Se copian los elementos de p12 a p13
    if ( p12->first != NULL ) {
        pn = p12->first;    // pn recorrerá p12
        pnodo = nodo_crear();
        pnodo->info = elemento_copiar(pn->info);
        if ( pnodo->info == NULL ) {
            nodo_liberar(pnodo);
            lista_liberar(p13);
            return NULL;
        }

        if ( pn3 == NULL ) { // p11 está vacía
            p13->first = pnodo;
            pn3 = pnodo;
        }
        else {
            pn3->next = pnodo;
            pn3 = pnodo;
        }
        while ( pn->next != NULL ) {
            pn = pn->next;
            pnodo = nodo_crear();
            pnodo->info = elemento_copiar(pn->info);
            if ( pnodo->info == NULL ) {
                nodo_liberar(pnodo);
                lista_liberar(p13);
                return NULL;
            }
            pn3->next = pnodo;
            pn3 = pnodo;
        }
    }

    return p13;
}

```