

# Recuperatorio Primer Parcial Laboratorio de computacion I

- Alumno: Zarate Enzo Tael
- División: 1C
- Legajo: 111262
- Mail: [tai.zarate9@gmail.com](mailto:tai.zarate9@gmail.com)

## **Biblioteca: Trabajo.h**

Función: void InicializarTrabajo (eTrabajo[], int);

Inicializa la variable isEmpty de todos los elementos del array, dándoles el valor -1.

Función: int BuscarLibreTrabajo (eTrabajo[], int);

Recorre el array de trabajos y busca un elemento que tenga su campo isEmpty con el valor -1, luego de esto, retorna el índice donde se encuentra ubicado este elemento.

Función: int AsignarIdTrabajo(eTrabajo[], int);

Se encarga de crear un id autoincremental y cada vez que se dé de alta un nuevo trabajo, esta le asigna el nuevo id verificando que el nuevo id creado se mayor al id de mayor valor numérico encontrado en el array.

Función: int BuscarIdTrabajo (int,eTrabajo[],int);

Verifica que el número que se le ingresó como parámetro exista como id de un trabajo, si existe retorna 1, si no existe retorna -1.

## **Biblioteca: Mascotas.h**

Función: void InicializarMascotas (eMascota[], int);

Inicializa la variable isEmpty de todos los elementos del array, dándoles el valor -1.

Función: int BuscarLibreMascotas (eMascota[], int);

Recorre el array de mascotas y busca un elemento que tenga su campo isEmpty con el valor -1, luego de esto, retorna el índice donde se encuentra ubicado este elemento.

Función: int BuscarIdMascotas (int,eMascota[], int);

Verifica que el número que se le ingresó como parámetro exista como id de una mascota, si existe retorna 1, si no existe retorna -1.

Función: void MostrarListadoMascotas (eMascota[], int);

Se encarga de mostrar todas las mascotas cargadas.

## **Biblioteca: Servicios.h**

Función: void InicializarServicio (eServicio[], int);

Inicializa la variable isEmpty de todos los elementos del array, dándoles el valor -1.

Función: int BuscarLibreServicio (eServicio[], int);

Recorre el array de servicios y busca un elemento que tenga su campo isEmpty con el valor -1, luego de esto, retorna el índice donde se encuentra ubicado este elemento.

Función: int BuscarIdServicios (int,eServicio[],int);

Verifica que el número que se le ingresó como parámetro exista como id de un servicio, si existe retorna 1, si no existe retorna -1.

Función: void MostrarListadoServicios (eServicio[], int);

Se encarga de mostrar todos los servicios cargados.

## **Biblioteca: Controlador.h**

Función: eTrabajo PedirTrabajo (eTrabajo[], int,eServicio[], int, eMascota[], int, eEmpleado[], int);

Se encarga de crear un nuevo trabajo y luego de que se le hayan cargado todos los valores a sus campos, la función retorna el nuevo trabajo con los campos cargados.

Función: int AltaTrabajo (eTrabajo[], int,eServicio[], int, eMascota[], int, eEmpleado[], int);

Da de alta un nuevo trabajo haciendo uso de la función eTrabajo PedirTrabajo (eTrabajo[], int,eServicio[], int, eMascota[], int, eEmpleado[], int) para cargar un trabajo.

Funcion: void BajaTrabajo (eTrabajo[], int,eServicio[], int, eMascota[], int, eEmpleado[], int);

La función pide que se ingrese un número, si el número coincide con el id de algún trabajo, el campo isEmpty de este trabajo pasa a estar con el valor -1, para que se considere que ese índice del array no tiene un elemento cargado.

Función: void ModificarTrabajo (eTrabajo[], int,eServicio[], int, eMascota[], int, eEmpleado[], int);

La función pide que se ingrese un número, si el número coincide con el id de algún trabajo, se permite cambiar el id del servicio, del empleado o de la mascota, a un trabajo.

Función: void MostrarListadoTrabajo (eTrabajo[], int,eServicio[], int, eMascota[], int, eEmpleado[], int);

Muestra todos los trabajos cargados hasta el momento.

La nueva estructura agregada es es la estructura eEmpleado:

typedef struct

```
{  
    int idEmpleado;  
    char nombre[120];  
    float salario;  
    int isEmpty;
```

```
} eEmpleado;
```

Esta se encarga de mostrar cuál es el empleado que tiene la responsabilidad de realizar el trabajo.

## **Biblioteca: Empleados.h**

Función: void inicializaEmpleados(eEmpleado empleados[], int tam\_empleados);

Inicializa la variable isEmpty de todos los elementos del array, dándoles el valor -1.

Función: int BuscarLibreEmpleado(eEmpleado empleados[], int tam\_empleados);

Recorre el array de empleados y busca un elemento que tenga su campo isEmpty con el valor -1, luego de esto, retorna el índice donde se encuentra ubicado este elemento.

Función: int BuscarIdEmpleado(int idIngresado,eEmpleado empleados[], int tam\_empleados);  
Verifica que el número que se le ingresó como parámetro exista como id de una empleado, si existe retorna 1, si no existe retorna -1.

Función: void mostrarTodosLosEmpleados(eEmpleado empleados[], int tam\_empleados);  
Muestra todos los empleados cargados.

Los nuevos informes agregados son:

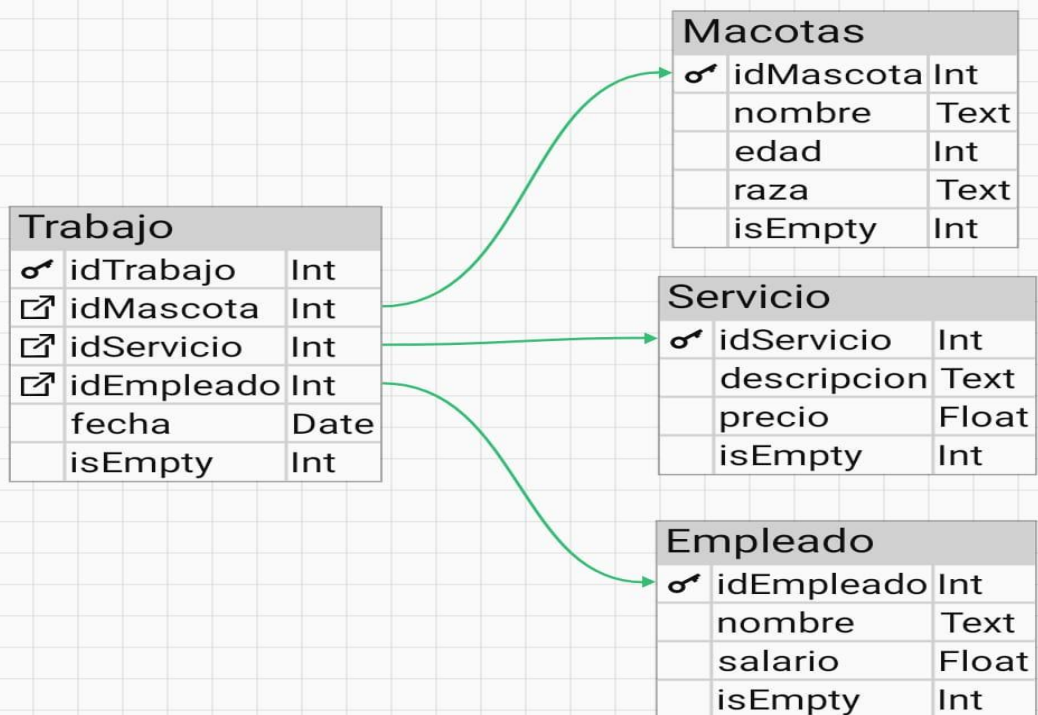
- Empleados con las mascotas que tienen a su cargo y el servicio que les corresponde a estas  
void empleadosResponsabilidades (eTrabajo[], int ,eServicio[], int , eMascota[], int , eEmpleado[], int );  
La función se encarga de recorrer el array de empleados, verificar de qué trabajos es responsable, luego busca en el trabajo, que servicio realizó y a que mascota. Por último muestra una lista donde se ven los nombres de cada empleado y qué mascota tiene a cargo, con el servicio que se le realizó.

- Mostrar el salario de los empleados, tanto el que le corresponde como el total considerando la cantidad y qué tipo de servicios realizó.

void MostrarPagaEmpleados(eTrabajo trabajos[], int tam, eServicio servicios[],int tam\_servicios, eEmpleado empleados[], int tam\_empleados);

La función se encarga de recorrer el array de empleados, mostrar el salario correspondiente de cada empleado y luego verifica si el empleado realizó algún trabajo, si lo hizo, se busca que tipo de servicio ha hecho, y se le suma al salario correspondiente un adicional que sería el costo del servicio realizado, si el

empleado no hizo ningún trabajo, se aclara que no ha hecho ningún trabajo como para tener un adicional.



Enlace: <https://drive.google.com/file/d/1687VQ8o-tFjh7pnebeYyYzQMBHjaEXAi/view?usp=sharing>