**“Año de la Universalización de la Salud”**

**UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS**



**FACULTAD DE INGENIERÍA**

**ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**Algoritmos y estructuras de datos**

**Profesor:**

Walter Cueva

**Proyecto:**

Apertura de establecimientos

**Integrantes:**

Sebastian Diaz Torres

**Ciclo:**

2020 - 1

**Caso de estudio:**

Ante la situación por la que está atravesando el Perú y el mundo se tiene que elegir que establecimiento pueden o no volver a sus labores normales luego de levantada la cuarentena.

Para ellos se dieron dos parámetros para saber que locales esta aptos para volver a brindar sus servicios.

**Parámetros:**

El carnet de desinfección que valida que el local tiene los medios para realizar una desinfección periódica del establecimiento.

Seguro para todos los trabajadores de la empresa en caso de contraer el virus.

Las estructuras que se usarán son:

listas enlazadas doble para organizar los establecimientos que deben ser organizados. Colas para los establecimientos permitidos y los establecimientos que no esta permitidos y tiene que regularizar sus permisos.

**Algoritmo principal.**

Encolar los establecimientos de la lista:

Lo que hace es recorrer la lista de establecimientos por organizar y según el criterio carnet de desinfección valido y el seguro para sus trabajadores en caso de contraer el virus.

Big0 = O(n)

void Recorre(Cola<Establecimientos> &Permitidos, Cola<Establecimientos> &NoPermitidos, Lista<Establecimientos> &NuevaLista)

{

  if (NuevaLista.GetLongitud() == 0) // 1

  {

    std::cout << "No hay locales en la lista" << std::endl; // 2

  }

  else{

    int cont = 0; //1

    Nodo<Establecimientos> \*aux = NuevaLista.inicio; //1

    while (cont < NuevaLista.GetLongitud()) //n

    {

      //Recorrido para meter en la cola a los locales con las caracteristicas

      EncolarLocales(aux->data, Permitidos, NoPermitidos);

      cont++;

      aux = aux->Siguiente;

    }

  }

}

void EncolarLocales(Establecimientos &Data, Cola<Establecimientos> &Permitidos, Cola<Establecimientos> &NoPermitidos)

{

  if (EntraEnLaCola(Data))

  {

    Permitidos.push(Data);

    std::cout << "Entro a la cola de permitidos \n";

    std::cout << Data.GetNombre() << std::endl;

  }

  else

  {

    NoPermitidos.push(Data);

    std::cout << "No esta en la cola de permitidos. \n";

    std::cout << Data.GetNombre() << std::endl;

  }

}

bool EntraEnLaCola(Establecimientos Data)

{

  //Condicion para que el establecimeinto entre o no a la cola de permitidos

  if (Data.GetSeguro() && Data.GetCarnet())

  {

    return true;

  }

  else

  {

    return false;

  }

}