|  |  |
| --- | --- |
| **Integrantes:** | *José Alejandro Gómez Castro ced- 207430945*  *Christian Ulloa Tosso ced- 109150113*  *Andrés Salas Olsen ced- 116510757* |

1. Propuesta del proyecto.

Una empresa de venta de anuncios a Canales de televisión, ha organizado las ventas llevando la información en una lista de Canales y una Lista de Anuncios de empresas, tomando en cuenta lo siguiente:

1. La lista de Canales de televisión es una lista doble enlazada de la información de los canales de transmisión, que contiene la siguiente información: Código del canal, Nombre del canal, teléfono, Monto mínimo a cobrar, tiempo máximo a transmitir, tiempo mínimo a transmitir, Costo por Minuto.
2. La lista de Anuncios a vender es una lista simple que contiene la siguiente información: Código de la empresa, nombre de la empresa, código del anuncio y tiempo de duración
3. Cada canal de la lista de Canales tiene una lista simple de AnunciosContratados, la cual tiene el código del anuncio, el costo total a Cobrar (con monto en 0 inicialmente) y un puntero al nodo del anuncio en la lista de Anuncios.

Realice lo siguiente:

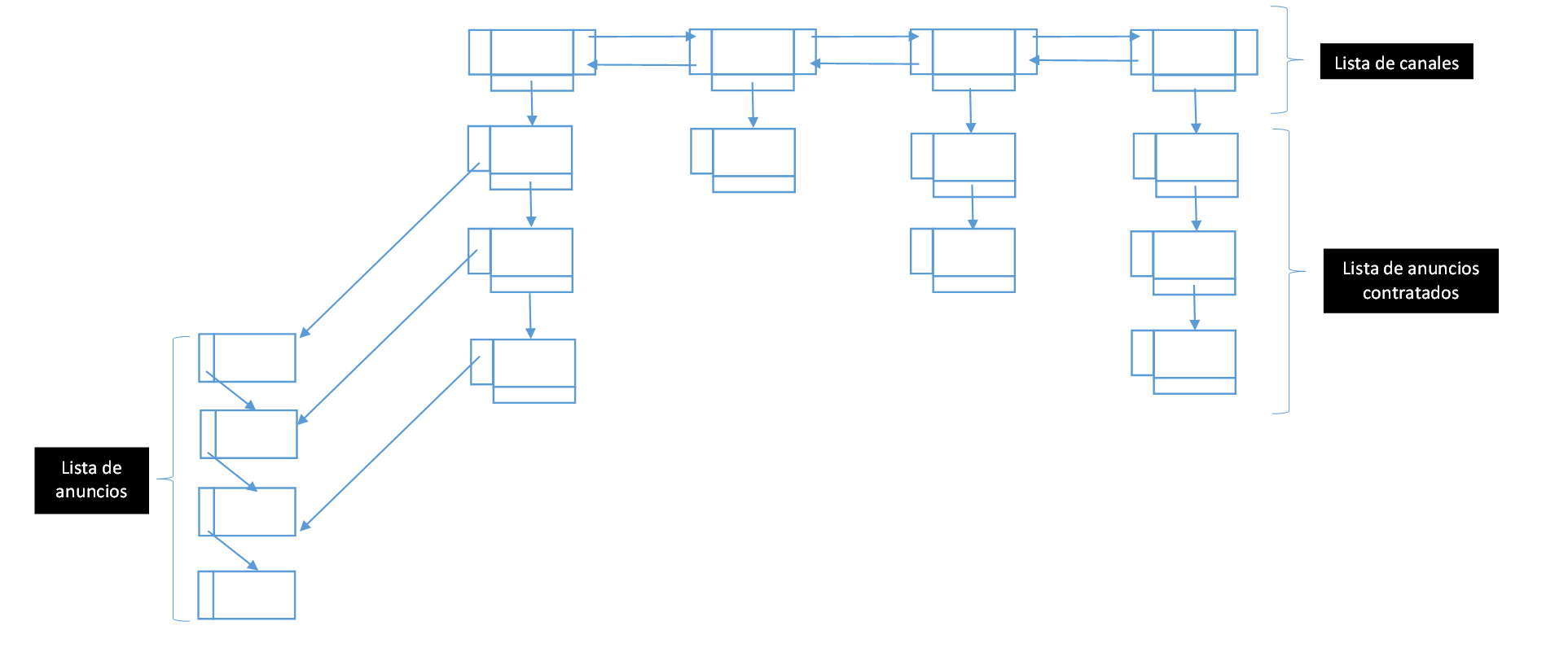
1. Defina el (los) TAD necesarios para dar solución a este problema en forma completa.
2. Haga el desarrollo de la solución utilizando C++, en la cual se incluya:

* Implemente: ActualizaCobros, donde actualice en la sublista de AnunciosContratados, el monto total a Cobrar, suponiendo que el Costo por minuto lo determina el Canal, y el tiempo de duración los Anuncios. Debe determinar:

1. Si un anuncio tiene menos tiempo en transmisión se le cobrará el Monto mínimo a Cobrar por el canal
2. Si un anuncio tiene mayor tiempo al máximo a trasmitir, se le cobrará el doble del Costo por minuto al doble, solo a los minutos extras.
3. Los parámetros serán la Lista de Canales y la Lista de Anuncios.

* Implemente ImprimeCanal, que imprima la información de Código de la empresa, nombre de la empresa código de anuncio y Monto a pagar, de todos los anuncios que serán contratados en un canal X. i. Los parámetros serán canal X, Lista de Canales y Lista de Anuncios.
* Implemente IncluirAnuncio que reciba como parámetro la lista de Canales, el canal X, la Lista de Anuncios, y el código de anuncio a incluir en un canal X. Debe incluir un nuevo nodo en la lista de AnunciosContratados del canal X, si el anuncio ya se encuentra incluido debe desplegar un mensaje y no incluirlo.

1. Desarrollo del Proyecto.
   1. Representación gráfica del Proyecto propuesto según listas y su relación.



* 1. Definición de TADs.

Para el desarrollo del Proyecto propuesto se definen los siguientes TAD:

TAD canalTV

**Representación abstracta:**

{ {1…99}, {a…z ̺ 1…9}, {####-####}, {+Q}, {N}, {N}, {+Q} }

**Invariantes:**

* **Código de canal**: Solamente puede ser un número del 1 al 99.
* **Nombre del canal**: Solamente puede estar formado por cualquier combinación de las letras del abecedario español, números del 1 al 9 y espacios.
* **Teléfono**: Solamente puede ser dos agrupaciones de 4 dígitos numéricos cada una.
* **Monto mínimo a cobrar**: Solamente puede ser un número Racional positivo.
* **Tiempo máximo a transmitir**: Solamente puede ser un número Natural.
* **Tiempo mínimo a transmitir**: Solamente puede ser un número Natural.
* **Costo por minuto**: Solamente puede ser un número Racional positivo.
* **Nodo anterior**: Solamente puede ser una referencia a un nodo del tipo TAD canalTV o Nulo.
* **Nodo siguiente**: Solamente puede ser una referencia a un nodo del tipo TAD canalTV o Nulo.
* **Nodo cabeza lista anuncios contratados**: Solamente puede ser una referencia a un nodo del tipo TAD anuncioContratado o Nulo.

**Operaciones y clasificación:**

**creaCanalTV**: racional, texto, texto, racional, natural, natural, racional, TAD Canal, TAD Canal, TAD AnuncioContratado->nada(constructora)

//Crea un nuevo canal

**ObtenerCodigoCanal**: nada->codigoCanal(analizadora)

//Obtiene el código del canal

**AsignarCodigoCanal**: racional->nada(modificadora)

//Asigna el código a un canal

**ObtenerTelefono**: nada->numeroTelefono(analizadora)

//Obtiene el número de teléfono

**AsignarTelefono**: texto->nada(modificadora)

//Asigna el número de teléfono

**ObtenerMontoMinimoCobro**: nada->Racional(analizadora)

//Obtiene el monto mínimo a cobrar por un anuncio.

**AsignarMontoMinimoCobro**: Racional->nada(modificadora)

//Asigna el monto mínimo a cobrar por un anuncio.

**ObtenerTiempoMaximoTransmitir**: nada->Natural(analizadora)

//Obtiene la cantidad de segundos máxima para la transmisión de un anuncio.

**AsignarTiempoMaximoTransmitir**: Natural->nada(modificadora)

//Asigna la cantidad de segundos máxima para la transmisión de un anuncio.

**ObtenerTiempoMinimoTransmitir**: nada->Natural(analizadora)

//Obtiene la cantidad de segundos mínima para la transmisión de un anuncio.

**AsignarTiempoMinimoTransmitir**: Natural->nada(modificadora)

//Asigna la cantidad de segundos mínima para la transmisión de un anuncio.

**ObtenerCostoPorMinuto**: nada->Racional(analizadora)

//Obtiene el costo mínimo por minuto (60 segundos).

**AsignarCostoPorMinuto**: Natural->nada(modificadora)

//Asigna el costo mínimo por minuto (60 segundos).

**ObtenerCostoPorMinuto**: nada->Racional(analizadora)

//Obtiene el costo mínimo por minuto (60 segundos).

**AsignarCostoPorMinuto**: Natural->nada(modificadora)

//Asigna el costo mínimo por minuto (60 segundos).

**ObtenerNodoAnterior**: nada->REF. TAD CanalTV(analizadora)

//Obtiene la dirección del canal anterior de la lista.

**AsignarNodoAnterior**: REF. TAD CanalTV ->nada(modificadora)

//Asigna la dirección del canal anterior de la lista.

**ObtenerNodoSiguiente**: nada->REF. TAD CanalTV(analizadora)

//Obtiene la dirección del canal siguiente de la lista.

**AsignarNodoSiguiente**: REF. TAD CanalTV ->nada(modificadora)

//Asigna la dirección del canal siguiente de la lista.

**ObtenerNodoAnuncioContratado**: nada->REF. TAD AnuncioContratado(analizadora)

//Obtiene la dirección del último anuncio contratado.

**AsignarNodoAnuncioContratado**: REF. TAD AnuncioContratado ->nada(modificadora)

//Asigna la dirección del último anuncio contratado.

**MostrarInformacion**: nada ->datos del TAD CanalTV(Salida en pantalla)

//Muestra los datos contenidos dentro del TAD CanalTV.

**Manejo de errores:**

Para el TAD CanalTV la verificación y correcta inclusión de los datos queda bajo la responsabilidad del usuario, siendo este el que a su vez es consciente de los tipos de dato que debe incluir según se le solicite, de acuerdo a los invariantes que presenta cada TAD. En este caso debe conocer los invariantes para el TAD CanalTV.

TAD anuncio

**Representación abstracta:**

{ {N}, {a…z ̺ 1…9}, {N}, {N} }

**Invariantes:**

**Código de empresa**: Solamente puede ser un número Natural.

**Nombre de la empresa**: Solamente puede estar formado por cualquier combinación de las letras del abecedario español, números del 1 al 9 y espacios.

**Código del anuncio**: Solamente puede ser un número Natural.

**Tiempo de duración**: Solamente puede ser un número Natural.

**Nodo siguiente**: Solamente puede ser una referencia a un nodo de tipo TAD anuncio o Nulo

**Operaciones y clasificación:**

**creaAnuncio**: Natural, texto, Natural, Natural, TAD Anuncio->nada(constructora)

//Crea un nuevo anuncio

**ObtenerCodigoEmpresa**: nada->codigoEmpresa(analizadora)

//Obtiene el código de la empresa del anuncio.

**AsignarCodigoEmpresa**: Natural->nada(modificadora)

//Asigna el código de la empresa del anuncio.

**ObtenerNombreEmpresa**: nada->nombreEmpresa(analizadora)

//Obtiene el nombre de la empresa del anuncio.

**AsignarNombreEmpresa**: Texto->nada(modificadora)

//Asigna el nombre de la empresa del anuncio.

**ObtenerCodigoAnuncio**: nada->CodigoAnuncio(analizadora)

//Obtiene el código del anuncio.

**AsignarCodigoAnuncio**: Natural->nada(modificadora)

//Asigna el código del anuncio.

**ObtenerTiempoDuracion**: nada->Natural(analizadora)

//Obtiene el tiempo de duración del anuncio.

**AsignarTiempoDuracion**: Natural->nada(modificadora)

//Asigna el tiempo de duración del anuncio.

**ObtenerNodoSiguienteAnuncio**: nada->REF.TAD Anuncio(analizadora)

//Obtiene la referencia al TAD Anuncio del siguiente anuncio.

**AsignarNodoSiguienteAnuncio**: REF.TAD Anuncio ->nada(modificadora)

//Asigna la referencia al TAD Anuncio del siguiente anuncio.

**MostrarInformacion**: nada ->datos del TAD Anuncio(Salida en pantalla)

//Muestra los datos contenidos dentro del TAD Anuncio.

**Manejo de errores:**

Para el TAD Anuncio la verificación y correcta inclusión de los datos queda bajo la responsabilidad del usuario, siendo este el que a su vez es consciente de los tipos de dato que debe incluir según se le solicite, de acuerdo a los invariantes que presenta cada TAD. En este caso debe conocer los invariantes para el TAD Anuncio.

TAD anuncioContratado

**Representación abstracta:**

{ {N}, {+Q}, {N} }

**Invariantes:**

**Código de anuncio**: Solamente puede ser un número Natural.

**Costo total**: Solamente puede ser un número Racional positivo o Cero.

**Nodo de origen**: Solamente puede ser una referencia a un nodo de tipo TAD anuncio y cuyo número de anuncio sea congruente con el TAD al corriente.

**Nodo siguiente**: Solamente puede ser una referencia a un nodo de tipo TAD anuncioContratado o Nulo.

**Operaciones y clasificación:**

**creaAnuncioContratado**: Natural, Racional, TAD Anuncio,

TAD AnuncioContratado->nada(constructora)

//Crea un nuevo anuncio contratado

**ObtenerCodigoAnuncio**: nada->codigoAnuncio(analizadora)

//Obtiene el código del anuncio contratado.

**AsignarCodigoAnuncio**: Natural->nada(modificadora)

//Asigna el código del anuncio contratado.

**ObtenerCostoTotal**: nada->Racional(analizadora)

//Obtiene el costo total del anuncio contratado.

**AsignarCostoTotal**: Racional->nada(modificadora)

//Asigna el costo total del anuncio contratado.

**ObtenerNodoSiguienteAnuncioContratado**: nada->REF.TAD AnuncioContratado(analizadora)

//Obtiene la referencia al TAD AnuncioContratado siguiente al anuncio contratado actual.

**AsignarNodoSiguienteAnuncioContratado**: REF.TAD AnuncioContratado ->nada(modificadora)

//Asigna la referencia al TAD AnuncioContratado siguiente al anuncio contratado actual.

**ObtenerNodoOrigenAnuncioContratado**: nada->REF.TAD Anuncio (analizadora)

//Obtiene la referencia al TAD Anuncio origen del anuncio creado.

**AsignarNodoOrigenAnuncioContratado**: REF.TAD Anuncio ->nada(modificadora)

//Asigna la referencia al TAD Anuncio origen del anuncio creado.

**MostrarInformacion**: nada ->datos del TAD AnuncioContratado(Salida en pantalla)

//Muestra los datos contenidos dentro del TAD AnuncioContratado.

**Manejo de errores:**

Para el TAD AnuncioContratado la verificación y correcta inclusión de los datos queda bajo la responsabilidad del usuario, siendo este el que a su vez es consciente de los tipos de dato que debe incluir según se le solicite, de acuerdo a los invariantes que presenta cada TAD. En este caso debe conocer los invariantes para el TAD AnuncioContratado.

TAD listaCanales

**Representación abstracta:**

{ lista:{TAD canalTv, {lista:{TAD anuncioContratado} } }

**Invariantes:**

**Longitud:** Solamente puede ser un numero Natural o Cero.

**Nodo cabeza canal:** Solamente puede ser una referencia al último TAD canalTv agregado o nulo.

**Lista de anuncios:** Solamente puede ser una referencia al TAD ListaAnuncios o nulo.

**Operaciones y clasificación:**

**creaListaCanales**: Natural, TAD CanalTv,->nada(constructora)

//Crea una nueva lista de canales

**ObtenerLongitud**: nada->longitudActual(analizadora)

//Obtiene la cantidad de canales de la lista.

**AsignarLongitud**: Natural->nada(modificadora)

//Asigna la cantidad de canales de la lista.

**ObtenerNodoCabeza**: nada->REF TAD CanalTV(analizadora)

//Obtiene la referencia al último nodo TAD CanalTV Agregado.

**AsignarNodoCabeza**: REF TAD CanalTV ->nada(modificadora)

//Asigna la referencia al último nodo TAD CanalTV Agregado.

**ObtenerListaAnuncios**: nada->REF TAD ListaAnuncios(analizadora)

//Obtiene la referencia a la lista de anuncios existente.

**AsignarListaAnuncios**: REF TAD ListaAnuncios ->nada(modificadora)

//Asigna la referencia a la lista de anuncios existente.

**AgregarCanal**: TAD CanalTV->nada(modificadora)

//Agregar nuevo canal a la lista de canales.

**ObtenerCanal**: Natural->TAD CanalTV(analizadora)

//Obtener nodo de canal de televisión.

**ObtenerAnuncio**: Natural->TAD CanalTV(analizadora)

//Obtener nodo de canal de televisión.

**AgregarContrato**: Natural, Natural->Booleano(modificadora)

//Agrega un contrato de anuncio de TV.

**ActualizarCobro**: nada->Lista de canales(analizadora)

//Actualizar el monto que se cobra por cada anuncio contratado en todos los canales.

**BuscarAnuncioContratado**: TAD CanalTV, TAD AnuncioContratado->booleano(analizadora)

//Actualizar el monto que se cobra por cada anuncio contratado en todos los canales.

**MostrarInformacion**: nada ->lista de MostrarInformacion TAD CanalTV, lista MostrarInformacion TAD AnunciosContratados(Salida en pantalla)

//Muestra los datos contenidos dentro de la lista.

**Manejo de errores:**

Para el TAD ListaCanales la verificación y correcta inclusión de los datos queda bajo la responsabilidad del usuario, siendo este el que a su vez es consciente de los tipos de dato que debe incluir según se le solicite, de acuerdo a los invariantes que presenta cada TAD. En este caso debe conocer los invariantes para el TAD ListaCanales.

Queda bajo la responsabilidad del usuario el no agregar dos canales con el mismo código.

Al tiempo queda bajo la responsabilidad del usuario el no agregar dos anuncios contratados con el mismo código en un mismo canal, en tal caso el sistema comunicará el caso y no permitirá la inclusión.

TAD listaAnuncios

**Representación abstracta:**

{ lista:{TAD anuncio} }

**Invariantes:**

**Nodo cabeza anuncio:** Solamente puede ser una referencia al último nodo agregado.

**Longitud:** Solamente puede ser un número Natural o Cero.

**Operaciones y clasificación:**

**creaListaAnuncios**: TAD Anuncio,Natural->nada(constructora)

//Crea una nueva lista de anuncios

**ObtenerLongitud**: nada->longitudActual(analizadora)

//Obtiene la cantidad de anuncios agregados en la lista.

**AsignarLongitud**: Natural->nada(modificadora)

//Asigna la cantidad de anuncios de la lista.

**ObtenerNodoCabeza**: nada->REF TAD Anuncio(analizadora)

//Obtiene el último TAD Anuncio agregado.

**AsignarNodoCabeza**: REF TAD Anuncio->nada(modificadora)

//Asigna el último TAD Anuncio agregado.

**MostrarInformacion**: nada ->lista de MostrarInformacion TAD Anuncios(Salida en pantalla)

//Muestra los datos contenidos dentro de la lista.

**Manejo de errores:**

Para el TAD ListaAnuncios la verificación y correcta inclusión de los datos queda bajo la responsabilidad del usuario, siendo este el que a su vez es consciente de los tipos de dato que debe incluir según se le solicite, de acuerdo a los invariantes que presenta cada TAD. En este caso debe conocer los invariantes para el TAD ListaAnuncios.

Queda bajo la responsabilidad del usuario el no agregar dos anuncios con el mismo código.