CS1112: Programación 2

Unidad 6: Relaciones entre clases

Sesión de Teoría - 10

Profesor:
José Antonio Fiestas Iquira jfiestas@utec.edu.pe

Material elaborado por:
Maria Hilda Bermejo, José Fiestas, Rubén Rivas

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Índice:

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Logro de la sesión:

Al finalizar la sesión, los alumnos diseñan e implementan Programas Orientados a Objeto, comprendiendo y utilizando los conceptos de asociación, composición y agregación.

Resumen clase 9

Programación Orientada a Objetos

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Como se define una clase en C++

Robot

Nombre
Color(es)
Tamaño
Vidas

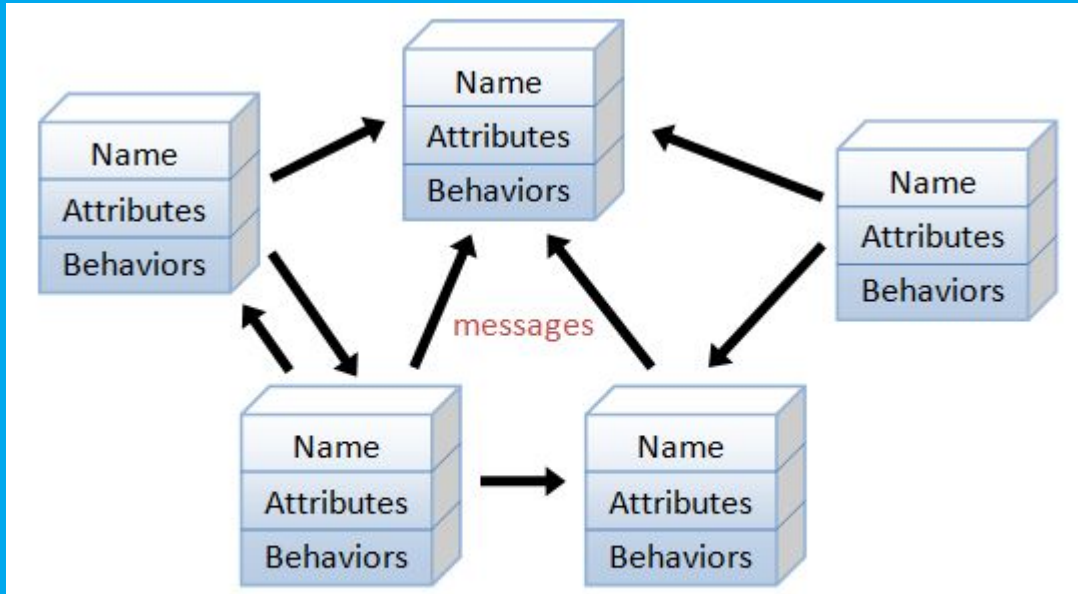
Moverse
Atacar
Girar
Morir

```
class CRobot {
 private:
    string m_Nombre;
    string m_Color;
    int    m_Tamanio;
    int    m_Vidas
    void   Girar();
 public:
   float m_PosX;
   float m_PosY;
   void  Moverse();
   void  Atacar();
};
```

Con acceso restringido

Con acceso libre

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA



An object-oriented program consists of many well-encapsulated objects and interacting with each other by sending messages

Relaciones entre clases

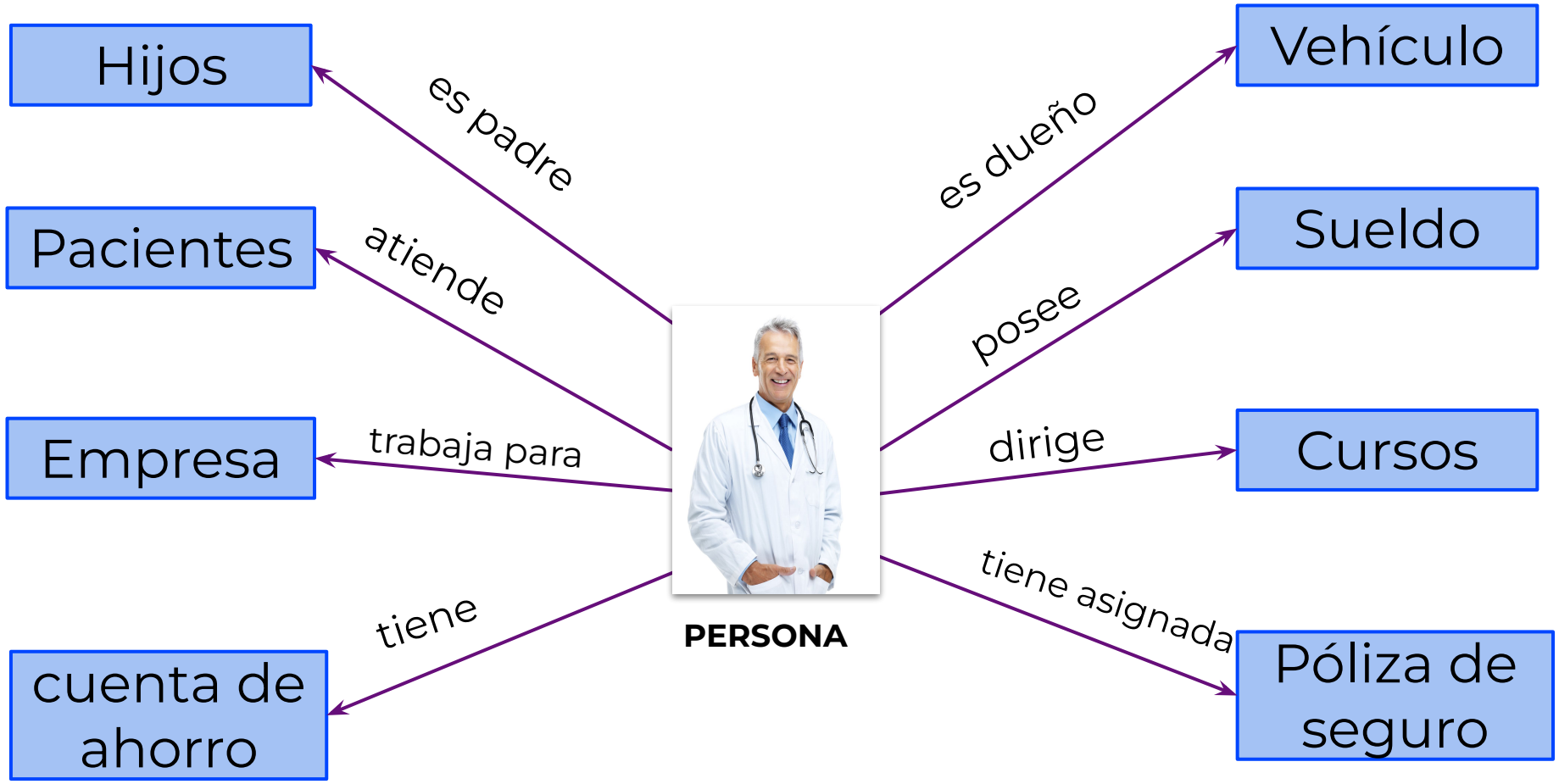
Definición de relaciones

Una relación es un vínculo entre dos objetos, este vínculo, se presenta por que, ambos objetos, se necesitan mutuamente para lograr la realización de una actividad o la realización de unos servicios.

Estas relaciones nacen por la misma dinámica social o por que la naturaleza así lo estableció.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Definición de relaciones



Tipos de relaciones entre clases

- Composición
- Agregación
- Asociación
- Herencia

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

6.1

Unidad 6: Relaciones entre clases

- Asociación

Definición de asociación

“ el momento en que dos objetos se unen para trabajar juntos y así, alcanzar una meta.”

Definición de asociación

“ Para validar la asociación, la frase **“Usa un”**, debe tener sentido:”

- El ingeniero usa una computadora
- El cliente usa una tarjeta de crédito

¿Se puede validar?

Para validar la asociación debemos ocupar la frase:

“ USA UN ”

- El ingeniero usa una computadora
- El cliente usa tarjeta de crédito

Ejemplo 1

Cliente - Tarjeta Credito

Se puede **asignar** o **retirar** la tarjeta de crédito,
sin que la existencia del Cliente sea vea afectada.

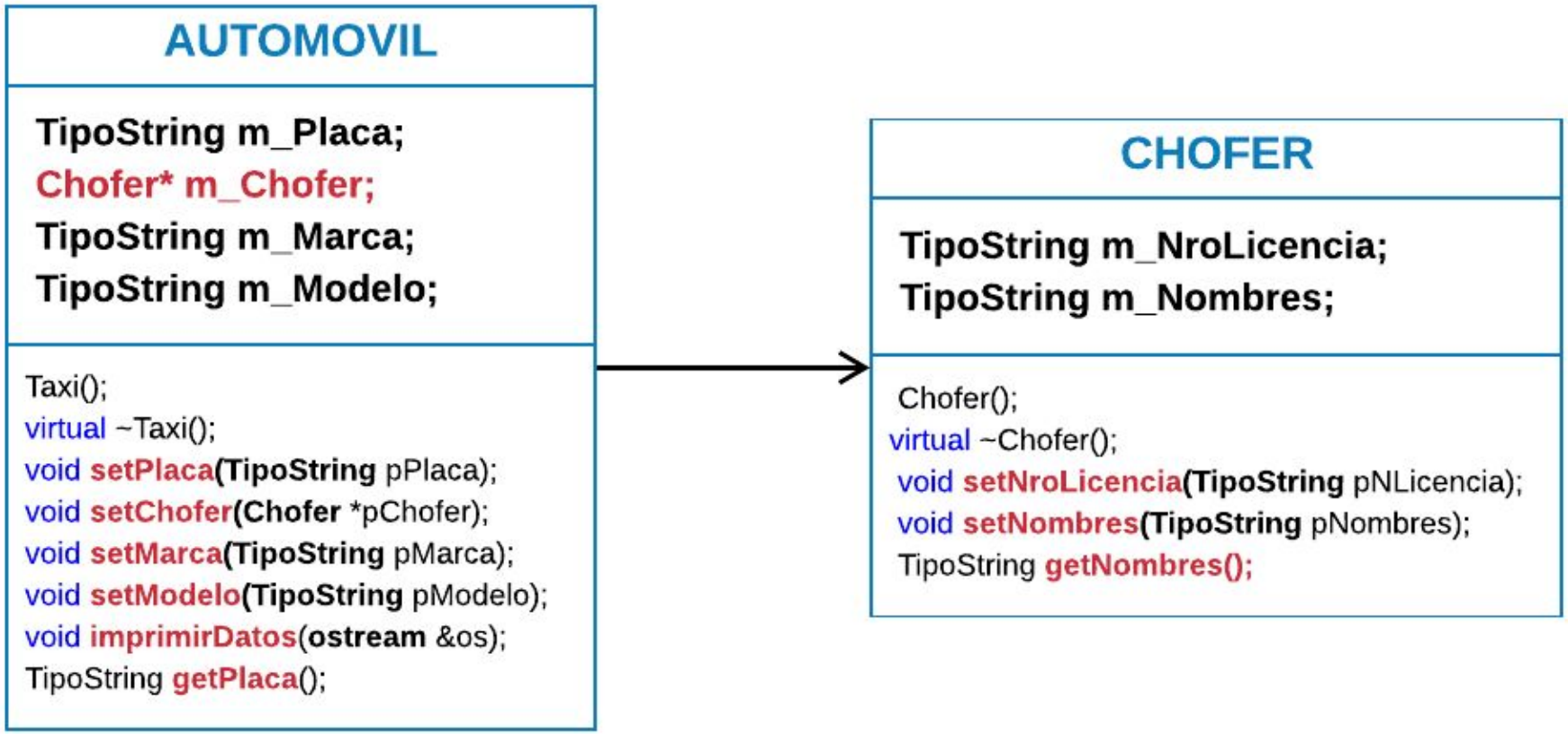
Ejemplo 2

Taxi - Chofer

Cada taxi **necesita** ser conducido por un chofer.

Notación

Es una línea con una flecha en el extremo apuntando a la clase asociada.



Ejemplo de asociación en C++

Tipos.h

```
#include <iostream>
using namespace std;

typedef int TipoEntero;
typedef char TipoCaracter ;
typedef string TipoString;
```

Ejemplo de asociación en C++

Taxi.h

```
#include "Tipos.h"
#include "Chofer.h"
class Taxi {
 private:
   TipoString m_Placa;
   Chofer* m_Chofer;
   TipoString m_Marca;
 public:
   Taxi();
   virtual ~Taxi();
   void setPlaca(TipoString pPlaca);
   void setChofer(Chofer *pChofer);
   void setMarca(TipoString pMarca);
   void imprimirDatos(ostream &os);
   TipoString getPlaca();
};
```

Chofer.h

```
#include "Tipos.h"
class Chofer {
 private:
 TipoString m_NroLicencia;
 TipoString m_Nombres;
 public:
  Chofer();
 virtual ~Chofer();
 void setNroLicencia(TipoString
pNLicencia);
 void setNombres(TipoString
pNombres);
 TipoString getNombres();
 };
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Ejemplo de asociación en C++

Taxi.cpp

```
#include "Taxi.h"
Taxi::Taxi(){}
Taxi::~Taxi(){}
void Taxi::setPlaca(TipoString pPlaca){
  m_Placa = pPlaca;
}
void Taxi::setChofer(Chofer *pChofer){
  m_Chofer = pChofer;
}
void Taxi::setMarca(TipoString pMarca){
  m_Marca = pMarca;
}
void Taxi::imprimirDatos(ostream &os){
  os << "Información del Taxi\n";
  os << "---------------------\n";
  os << "Placa : " << m_Placa << endl;
  os << "Chofer : " <<
m_Chofer->getNombres();
}
TipoString Taxi::getPlaca(){
  return m_Placa;
}
```

Chofer.cpp

```
#include "Chofer.h"
 Chofer::Chofer(){}
 Chofer::~Chofer(){}
 void Chofer::setNroLicencia(TipoString
pNLicencia){
    m_NroLicencia=pNLicencia;
 }
 void Chofer::setNombres(TipoString
pNombres){
    m_Nombres = pNombres;
 }
TipoString  Chofer::getNombres(){
    return m_Nombres;
 }
```

Ejemplo de asociación en C++

main.cpp

```
#include "Chofer.h"
#include "Taxi.h"
int main() {
  Chofer * pJuan = new Chofer();
  Taxi * pToyota = new Taxi();
  pJuan ->setNombres("Juan Perez");
  pToyota->setPlaca("A1Z521");
  pToyota->setChofer(pJuan);
  pToyota->imprimirDatos(cout);
  cout<<"HOLA";
  delete pJuan ;
  delete pToyota;
}
```

console.cpp

```
> ./main
Información del Taxi
------------------------
Placa : A1Z521
Chofer : Juan Perez
```

Fuente completo en repl.it: **http://bit.ly/2PkPvZu**

6.2

Unidad 6: Relaciones entre clases
- Composición

Definición de composición

"... es un tipo de relación **dependiente** en dónde un objeto más complejo es **conformado** por objetos más pequeños."

Definición de composición

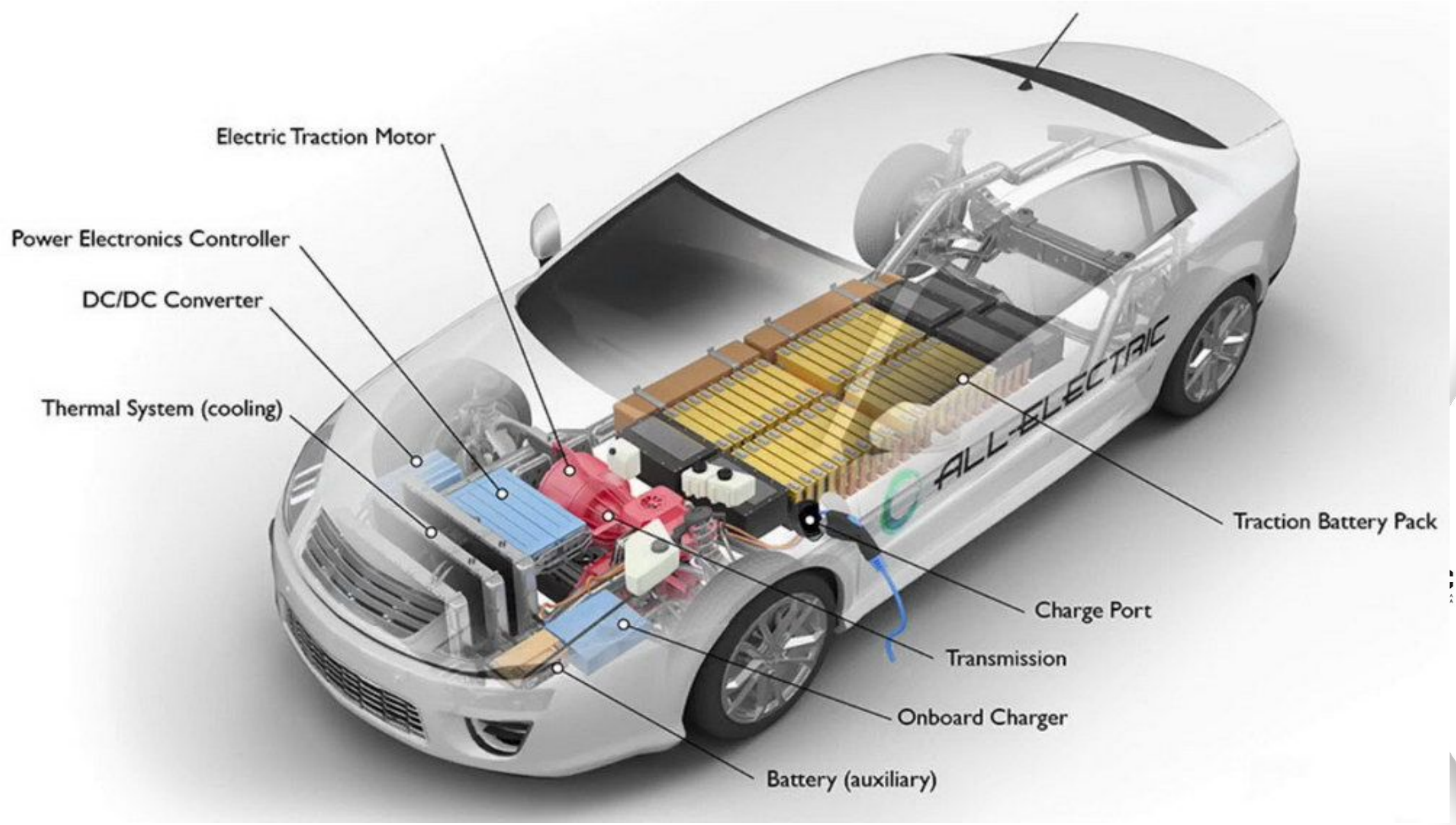
"... El objeto en el nivel superior de la jerarquía es el **todo** y los que están en los niveles inferiores son sus **partes o componentes"**

Definición de composición

"... Objetos componentes(Partes) como la clase contenedora, **nacen** y **mueren** al mismo tiempo."

>> Tiene el mismo tiempo de vida.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA


Electric Traction Motor
Power Electronics Controller
DC/DC Converter
Thermal System (cooling)
ALL-ELECTRIC
Traction Battery Pack
Charge Port
Transmission
Onboard Charger
Battery (auxiliary)

¿Se puede validar?

Para validar la composición debemos utilizar la frase:

“ Es parte de ”

- Un muro es parte de un usuario (facebook).
- Las llantas son partes del auto.
- El motor es parte del auto.

Ejemplo 1

Usuario - Muro

Al crear una cuenta de Usuario, se crea un muro para que el usuario publique.

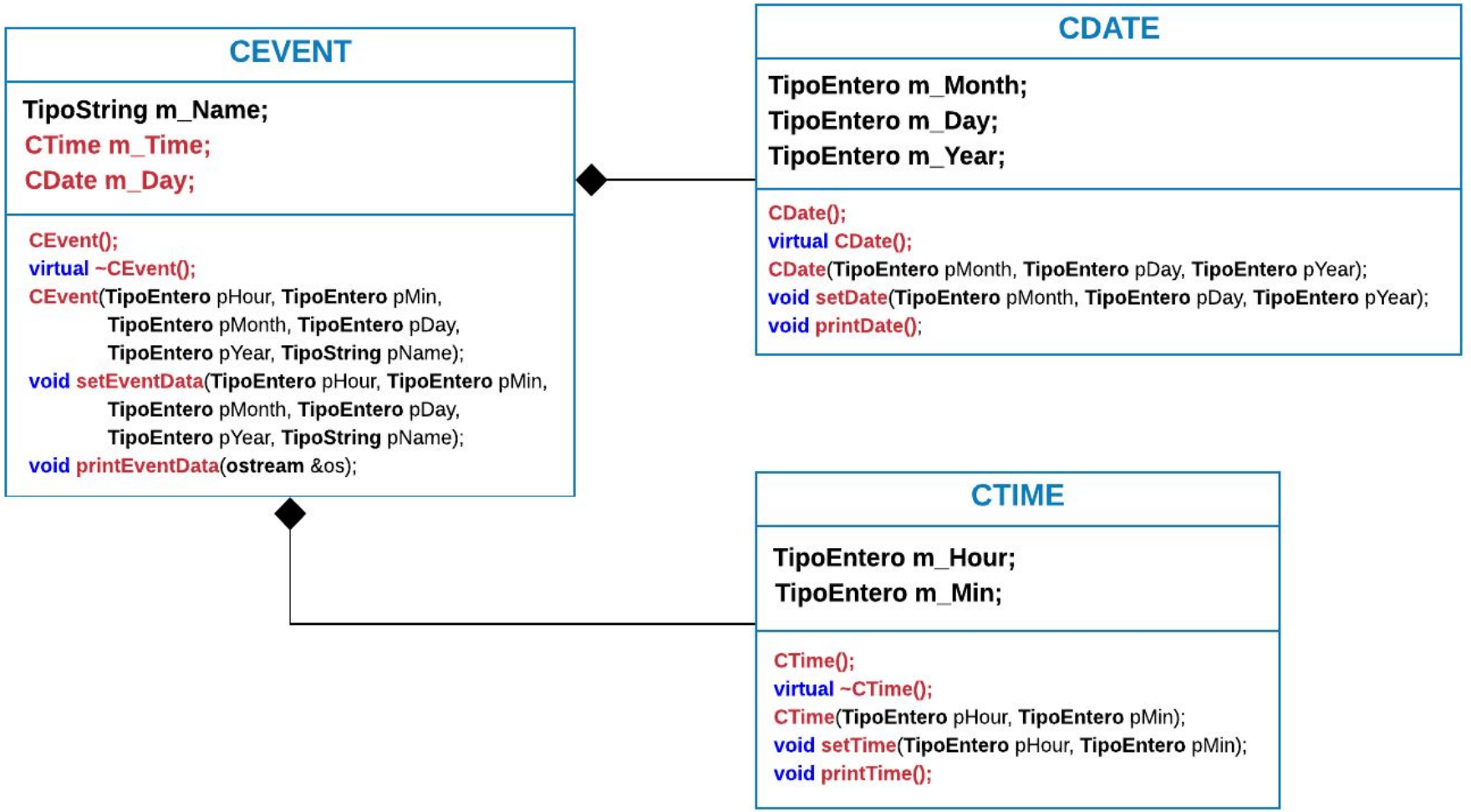
Ejemplo 2

Eventos - Fecha y Hora

Un evento está compuesto por una fecha y hora

Notación

Es una línea con un rombo lleno en el extremo apuntado a la clase **todo.**



Ejemplo de composición en C++

CDate.h

```
#include "Tipos.h"
class CDate {
private:
   TipoEntero m_Month;
   TipoEntero m_Day;
   TipoEntero m_Year;
public:
  Date();
  virtual ~ CDate();
 CDate(TipoEntero pMonth, TipoEntero pDay,
         TipoEntero pYear);
 void setDate(TipoEntero pMonth, TipoEntero
pDay,
                              TipoEntero
pYear);
 void printDate(ostream &os);
};
```

CTime.h

```
#include "Tipos.h"
class CTime {
 private:
   TipoEntero m_Hour;
   TipoEntero m_Min;
 public:
   CTime();
   virtual ~CTime();
   CTime(TipoEntero pHour, TipoEntero
pMin);
   void setTime(TipoEntero pHour,
TipoEntero pMin);
   void printTime(ostream &os);
};
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

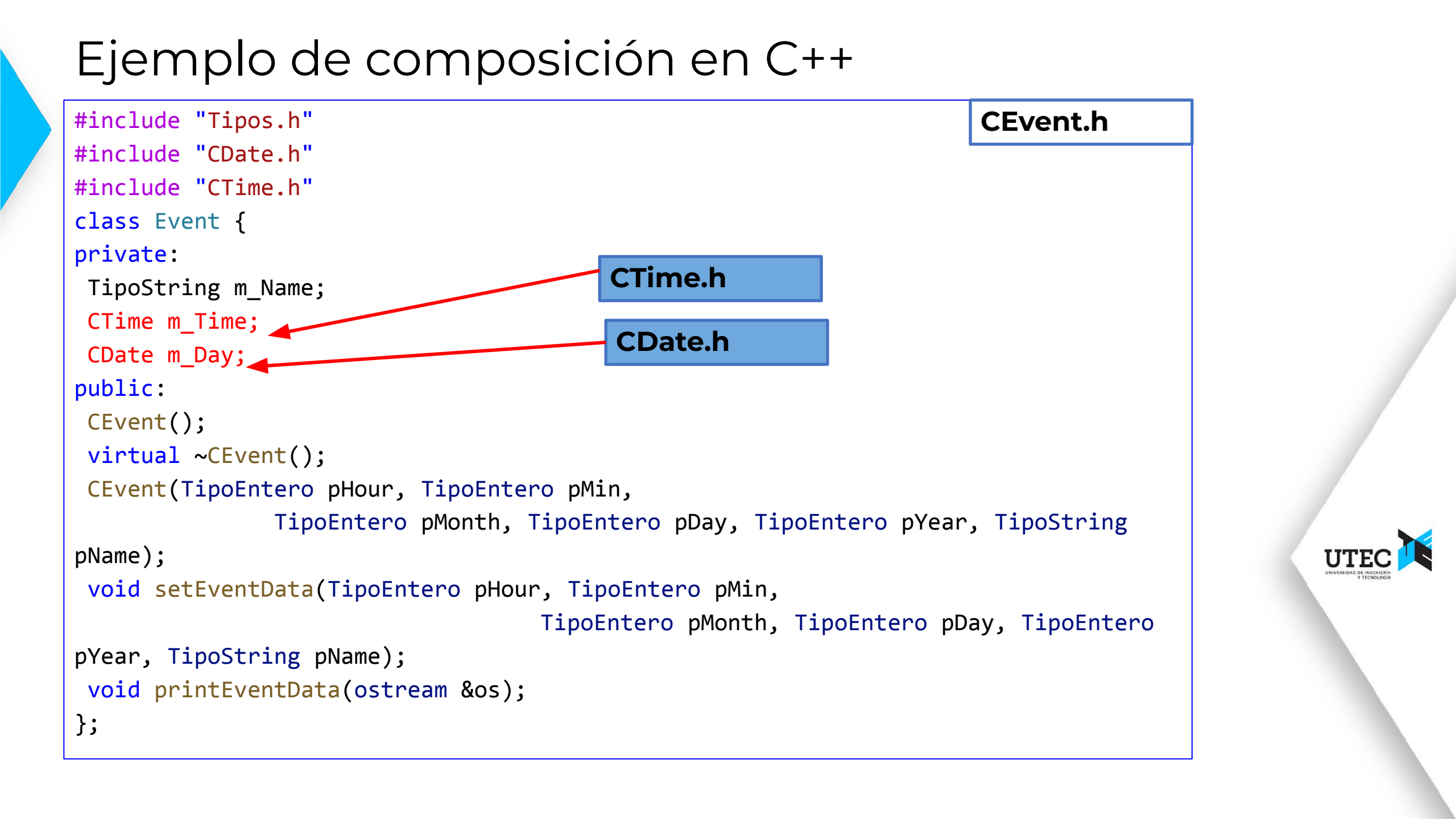
Ejemplo de composición en C++

CEvent.h

```
#include "Tipos.h"
#include "CDate.h"
#include "CTime.h"
class Event {
private:
 TipoString m_Name;
 CTime m_Time;
 CDate m_Day;
public:
 CEvent();
 virtual ~CEvent();
 CEvent(TipoEntero pHour, TipoEntero pMin,
                TipoEntero pMonth, TipoEntero pDay, TipoEntero pYear, TipoString
pName);
 void setEventData(TipoEntero pHour, TipoEntero pMin,
                                    TipoEntero pMonth, TipoEntero pDay, TipoEntero
pYear, TipoString pName);
 void printEventData(ostream &os);
};
```

CTime.h

CDate.h



Ejemplo de composición en C++

CTime.cpp

```
#include "CTime.h"
CTime::CTime(TipoEntero pHour, TipoEntero pMin) {
    setTime(pHour,pMin);
}
void CTime::setTime(TipoEntero pHour, TipoEntero pMin) {
     if ( 0 <= pHour && pHour < 24 )
           m_Hour = pHour ;
     else m_Hour = 0;
     if ( 0 <= pMin && pMin < 60 )
           m_Min = pMin ;
     else m_Min = 0;
}
void CTime::printTime(ostream &os) {
     os << setw(2) << setfill('0') << m_Hour << ":"
        << setw(2) << setfill('0') << m_Min;
 }
```

Constructor

Ejemplo de composición en C++

CDate.cpp

```
#include"CDate.h"
CDate::CDate() {
   m_Month= 1;    m_Day = 1;     m_Year = 1900;
}
CDate::CDate(TipoEntero  pMonth, TipoEntero pDay,   TipoEntero pYear) {
    if ( pMonth >= 1 && pMonth <= 12 )
    m_Month = pMonth;
    else m_Month = 1;
    if ( pDay >= 1 && pDay <= 31 )
    m_Day = pDay;
    else m_Day = 1;
    if ( pYear >= 1900 && pYear <= 2010 ) m_Year = pYear;
    else m_Year = 1900;
}
void CDate::printDate(ostream &os) {
   os << setw(2) << setfill('0') << m_Month << "/"
        << setw(2) << setfill('0') << m_Day << "/" << m_Year;
}
```

Ejemplo de composición en C++

CEvent.cpp

```
#include "CEvent.h"
CEvent::CEvent(){}
CEvent::~CEvent(){}

void CEvent::setEventData(TipoEntero  pHour, TipoEntero pMin, TipoEntero
pMonth, TipoEntero pDay, TipoEntero pYear, TipoString p
     m_Time.setTime(pHour, pMin);
     m_Day.setDate(pMonth, pDay, pYear);
     m_Name = pName ;
}
void CEvent::printEventData(ostream &os) {
     os << eventName << " es ";
     m_Day.printDate(os);
     os << " a las  ";
     m_Time.printTime(os);
     os << endl;
}
```

CTime.cpp

CDate.cpp

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Ejemplo de composición en C++

CEvent.cpp

```
#include "CEvent.h"
CEvent::CEvent(){}
CEvent::~CEvent(){}
CEvent::CEvent(TipoEntero  pHour, TipoEntero pMin, TipoEntero pMonth, TipoEntero
pDay,
TipoEntero pYear, TipoString pName )  : m_Time(pHour, pMin), m_Day(pMonth, pDay,
pYear)
{
      m_Name = pName ;
}
```

CTime.cpp

CDate.cpp

¿Y CÓMO SETEAMOS USANDO CONSTRUCTORES?

Ejemplo de composición en C++

main.cpp

```
#include "Event.h"


int main() {
 CEvent * pEvent = new CEvent();
 pEvent .setEventData(6, 0, 12, 25, 2019,
"Navidad");
 pEvent .printEventData(cout);
 CEvent * pEvent2 = new CEvent();
 pEvent2 .setEventData(1, 15, 13, 9, 2019,
"Open Day");
 pEvent2 .printEventData(cout);
}
```

console.cpp

```
cpp main.cpp
> ./main
Navidad es 12/25/2019 a las  06:00
Open Day es 01/09/2019 a las  01:15
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Ejemplo de composición en C++

main.cpp

```
#include "CEvent.h"
int main() {
 CEvent * pEvent = new CEvent(6, 0, 12, 25, 2019,
"Navidad");
 pEvent .printEventData(cout);
 CEvent * pEvent2 = new CEvent(1, 15, 13, 9, 2019, "Open
Day");
 pEvent2 .printEventData(cout);
}
```

console.cpp

```
cpp main.cpp
> ./main
Navidad es 12/25/2019 a las  06:00
Open Day es 01/09/2019 a las  01:15
```

¿Y COMO SETEAMOS USANDO CONSTRUCTORES?

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

6.3

Unidad 6: Relaciones entre clases

- Agregación

Definición de agregación

"... representa la relación "**es parte de**", una o más clases son partes de un conjunto. Es decir, relación entre el **todo** y sus **partes**, se representa agregando un diagrama vacío en el extremo que corresponde al todo".

Definición de agregación

"... Contiene un atributo, que puede ser una colección, es decir un array, vector, etc, y además de ello la clase que contiene la colección debe tener un método que agregue los elementos a la colección.

¿Se puede validar?

Para validar la agregación debemos ocupar la frase:

“ TIENE UN ”

- La universidad agrupa varios estudiantes
- Un estudiante tiene varios cursos

Ejemplo 1

Universidad - Estudiante

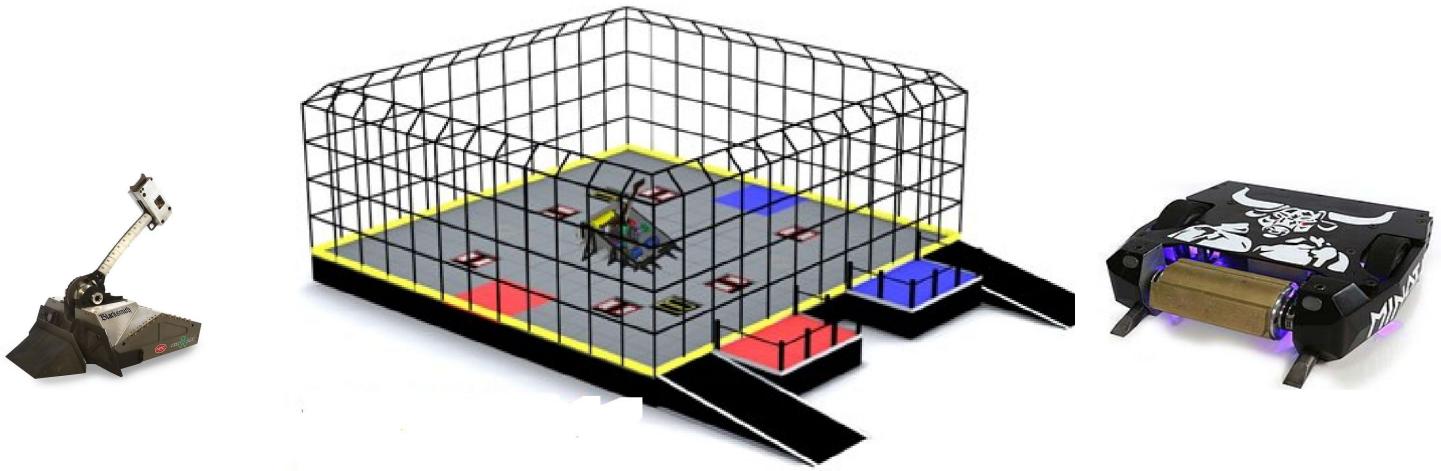
Universidad agrupa a varios estudiantes

Ejemplo 2

Estudiante - Cursos

El estudiante tiene varios cursos

Ejemplo 3

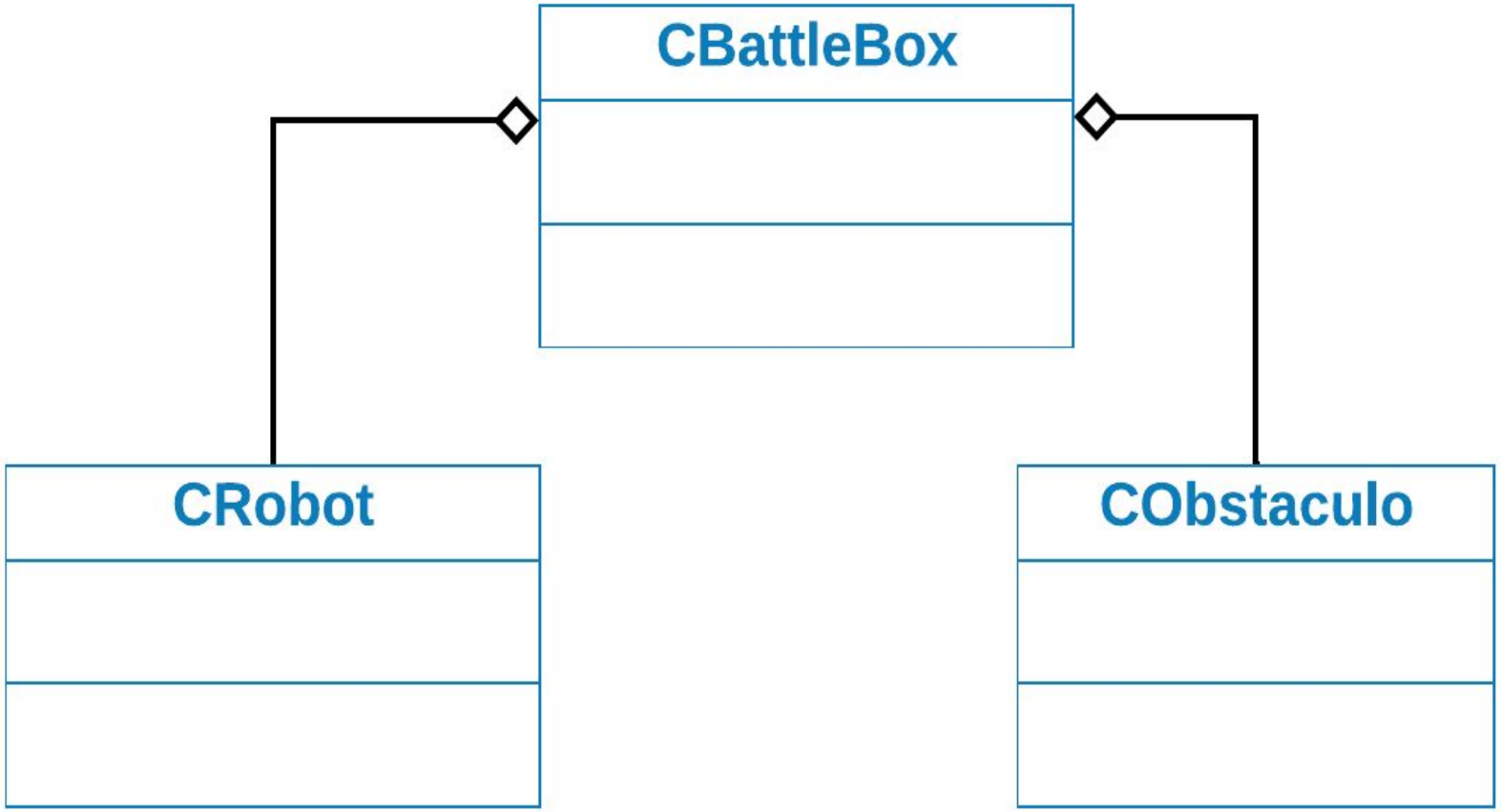


Battlebox - Robots y Obstáculos

Battlebox tiene Robos y Obstáculos

Notación

Es una línea con un rombo en el extremo apuntado a la clase **Contenedora.**



Ejemplo de relación de Agregación en C++

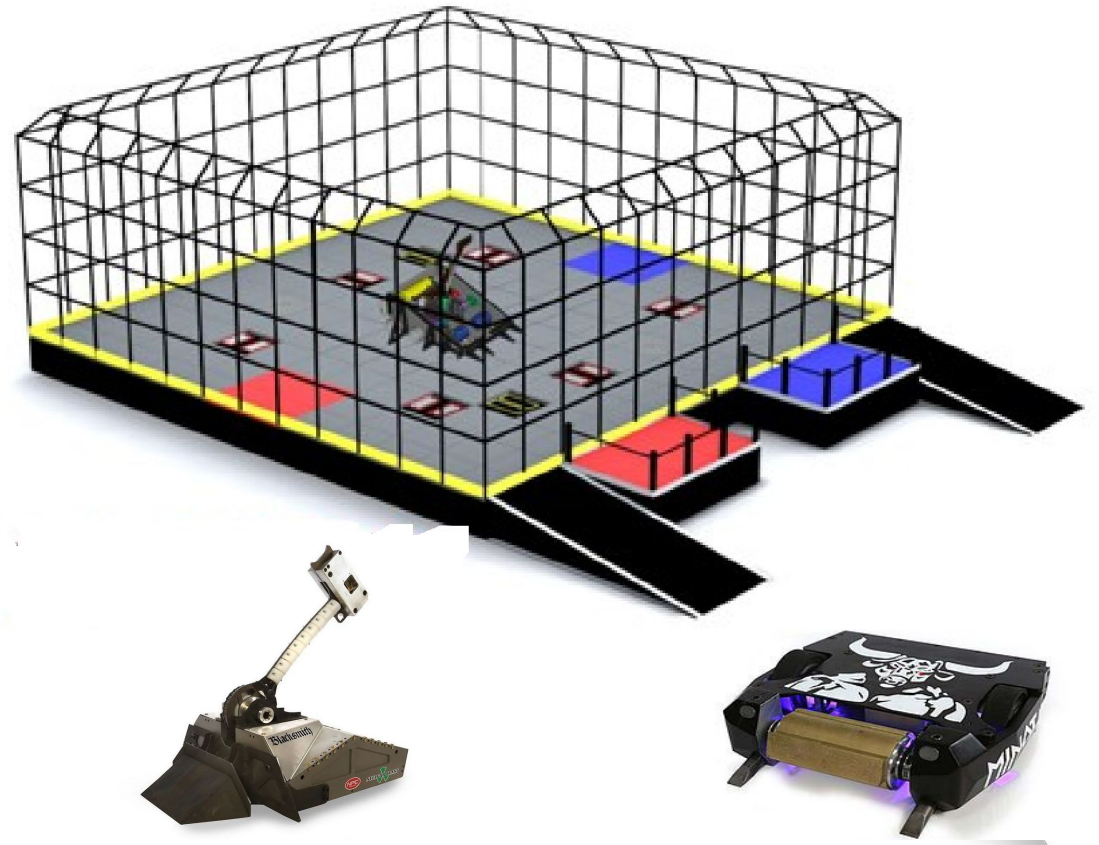
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Ejemplo con BattleBots

Ver video

¿Qué podemos abstraer?

- Objetos **Robots** (Bloodsport, blacksmith, Bronco)
- Objeto **Obstáculos** (Mazo Pulverizador, Tiras de Spike, Spinners, Kill Saws)
- Objeto **BattleBox** de tipo Matriz 48x48
- Ubicar Robots aleatoriamente en el **BattleBox**
- Generar Obstáculos aleatoriamente en el **BattleBox**



Clase CRobot

Definición de la clase **CRobot** para crear **instancias** de **robots:**

- Bloodsport,
- Blacksmith,
- BroncoMazo
- ....

CRobot

```
TipoString    m_Nombre;
TipoCaracter  m_Color;
TipoEntero    m_PosX;
TipoEntero    m_PosY;
TipoEntero m_Vidas;
```

```
CRobot();
CRobot(TipoString pNombre, TipoCaracter
pColor,TipoEntero pPosX, TipoEntero pPosY);
virtual ~CRobot();
void setNombre(TipoString pNombre);
void moverse(TipoEntero pPosX, TipoEntero pPosY);
void Chocar(CObstaculo &pObstaculo);
TipoString getNombre();
TipoEntero getPosX();
TipoEntero getPosY();
TipoCaracter getColor();
TipoString mostrarPosicion();
```

Clase CObstaculo

Definición de la clase **CObstaculo** para crear **instancias** de **obstáculos:**

- Mazo Pulverizador,
- Tiras de Spike,
- Spinners,
- Kill Saws

....

CObstaculo

TipoString m_Nombre;
TipoCaracter m_Color;
TipoEntero m_PosX;
TipoEntero m_PosY;
TipoEntero m_Daño;

CObstaculo();
CObstaculo(TipoString pNombre, **TipoCaracter** pColor,**TipoEntero** pPosX, **TipoEntero** pPosY);
virtual ~CObstaculo();
void setNombre(TipoString pNombre);
void moverse(TipoEntero pPosX, **TipoEntero** pPosY);
TipoString **getNombre();**
TipoEntero **getPosX();**
TipoEntero **getPosY();**
TipoCaracter **getColor();**
TipoString **mostrarPosicion();**

Clase CBattleBox

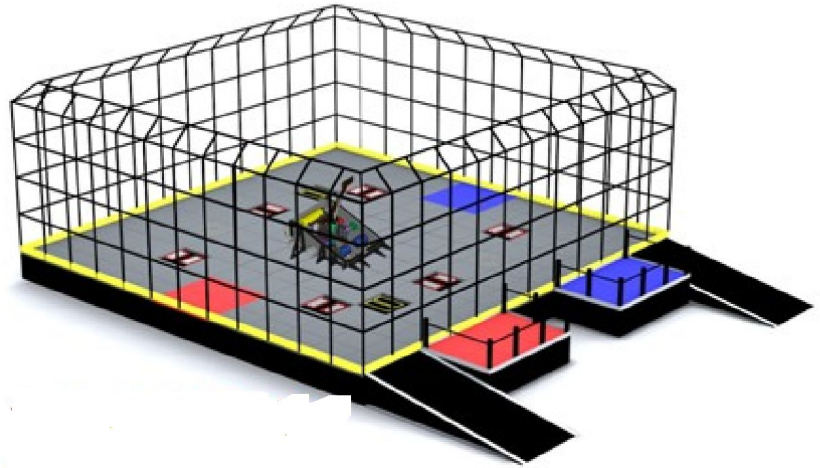
Definición de la clase **CBattleBox** para crear **agregación** de **Robots** y **Obstáculos** mediante el uso de **vector< ... >**

....

CBattleBox

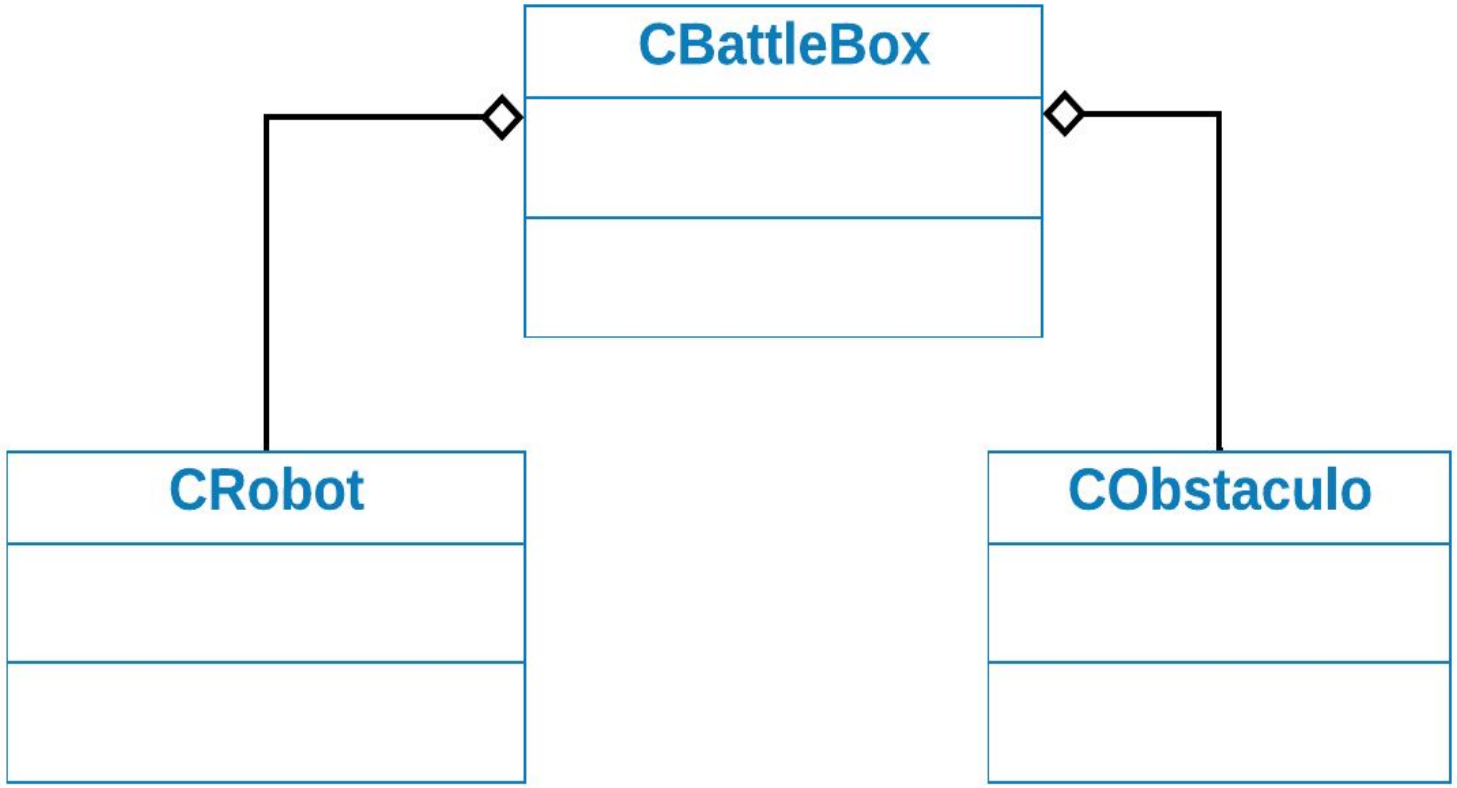
```
TipoEntero m_Altura;
TipoEntero m_Ancho;
TipoEntero m_CantRobots;
TipoEntero m_CantObstaculos;
vector<vector<TipoCaracter>> m_Plano;
vector<CRobot*> m_Robots;
vector<CObstaculo*> m_Obstaculos;
```

```
CBattleBox();
CBattleBox(TipoEntero pAltura, TipoEntero pAncho);
virtual ~CBattleBox();
void adicionarRobot(CRobot* pRobot);
CRobot* removerRobot(TipoString pNombre);
CRobot* buscarRobot(TipoString nombre);
void imprimirRobots();
TipoEntero getAltura();
TipoEntero getAncho();
TipoEntero getCantidadObjectos();
void dibujarBattleBox();
void actualizarBattleBox();
```



Formalizando con la Notación de Agregación

La clase **CBattleBox** contiene varios elementos del tipo **CRobot** y **CObstaculo**.



Formalizando con la Notación de Agregación

...Ahora las clases con sus **atributos** y **métodos,** y relacionados entre ellos.

Ejemplo:

El programa muestra el **BATTLEBOX** y permite agregar, mostrar a Robots y Obstáculos.
De cada robot, se conoce su nombre, su ubicación (coordenada x, y) y el color.

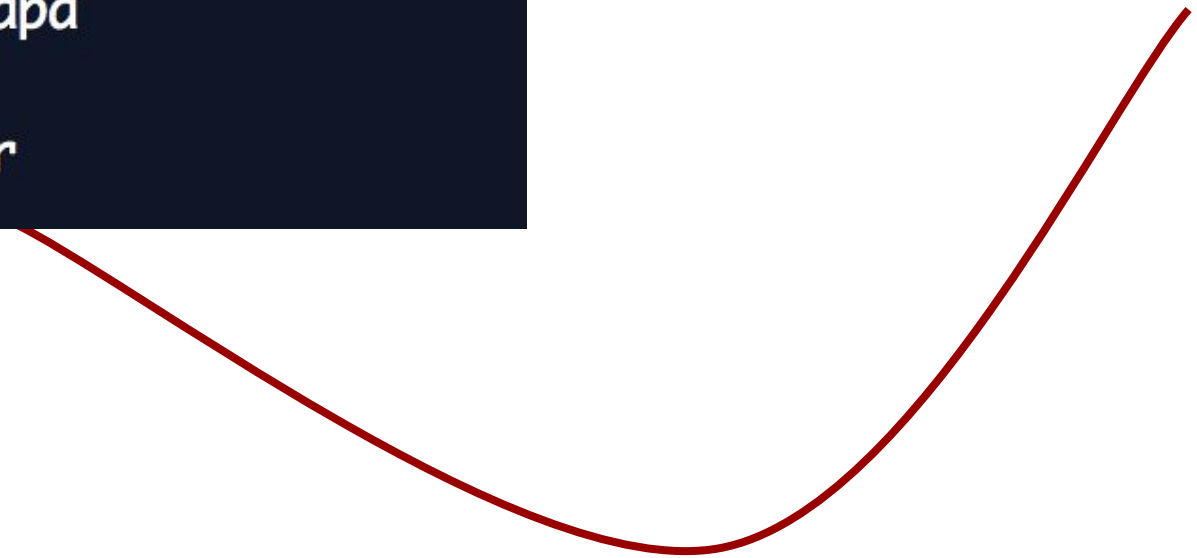
Ejemplo:

```
MENU
----

1. Agregar un nuevo objeto
2. Remover objeto
3. Dibujar Mapa

0. Para Salir
```

```
Ingrese Nombre: Bomblebee
Ingrese color (Un caracter): A
Ingrese posición X : 4
Ingrese posición Y : 7
```



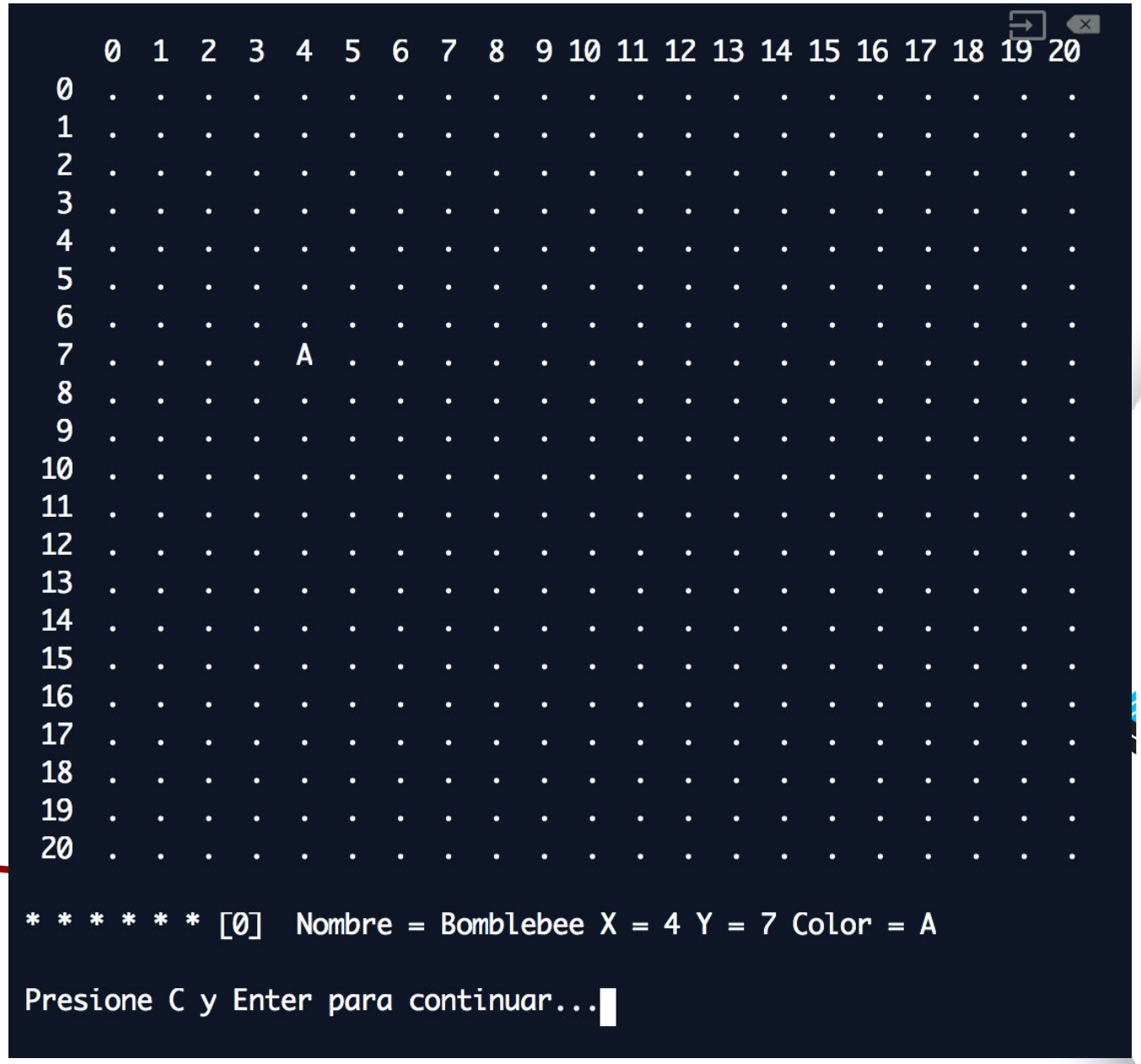
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Ejemplo:

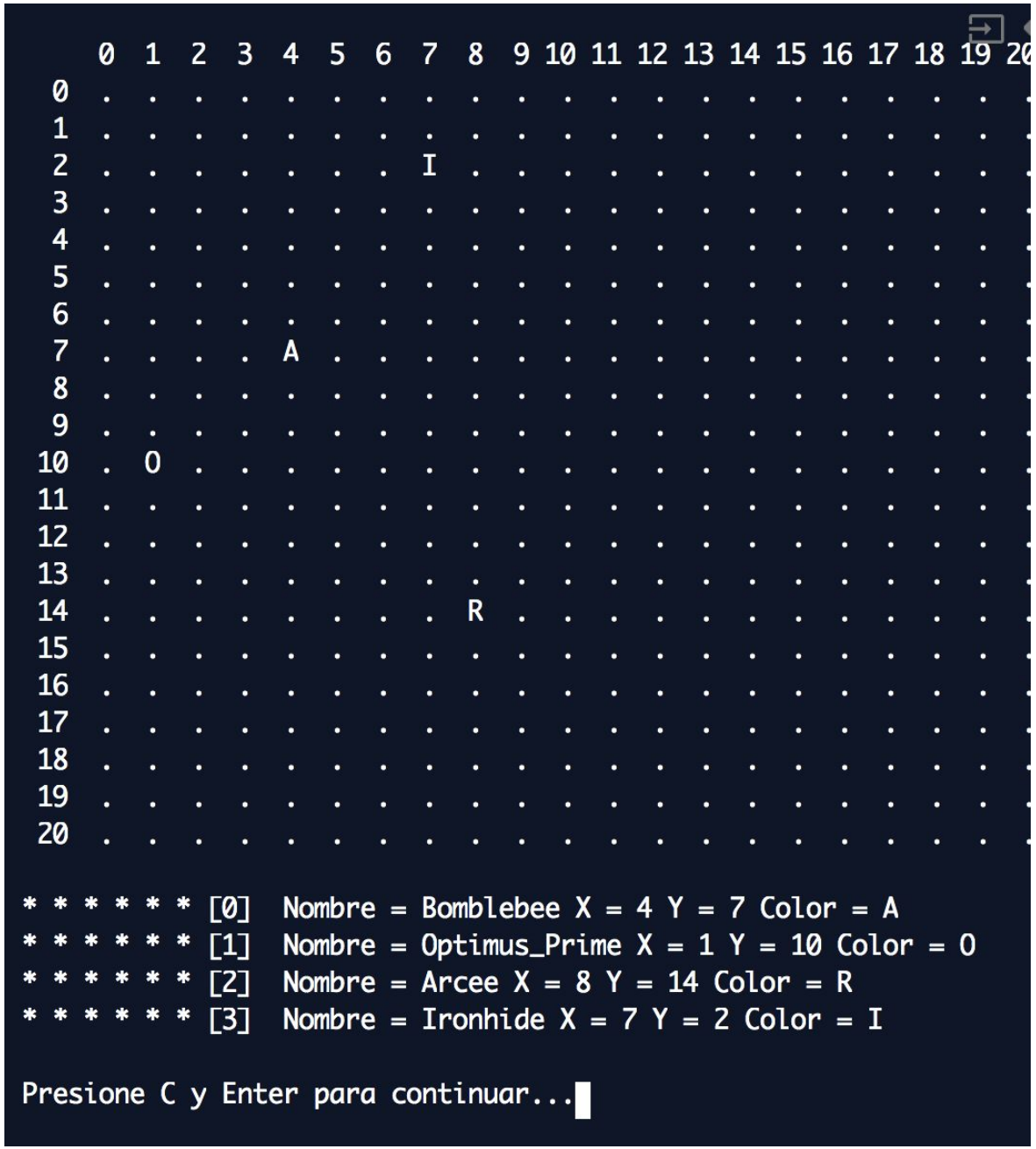
```
MENU
----

1. Agregar un nuevo objeto
2. Remover objeto
3. Dibujar Mapa

0. Para Salir
```



Ejemplo:


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
0 . . . .
1 . . . .
2 . . . . I . . . .
3 . . . .
4 . . . .
5 . . . .
6 . . . .
7 . . . . A . . . .
8 . . . .
9 . . . .
10 . O . . . .
11 . . . .
12 . . . .
13 . . . .
14 . . . . R . . . .
15 . . . .
16 . . . .
17 . . . .
18 . . . .
19 . . . .
20 . . . .
* * * * * * [0] Nombre = Bomblebee X = 4 Y = 7 Color = A
* * * * * * [1] Nombre = Optimus_Prime X = 1 Y = 10 Color = O
* * * * * * [2] Nombre = Arcee X = 8 Y = 14 Color = R
* * * * * * [3] Nombre = Ironhide X = 7 Y = 2 Color = I
Presione C y Enter para continuar...

Ejemplo de Agregación en C++ (BattleBots)

Tipos.h

```
#include <iostream>
using namespace std;

typedef int TipoEntero;
typedef char TipoCaracter ;
typedef string TipoString;
```

Ejemplo de Agregación en C++ (BattleBots)

CRobot.h

```
#include "TipoDato.h"
class CRobot {
 private:
   TipoString    m_Nombre;
   TipoCaracter  m_Color;
  void moverse(TipoEntero pPosX, TipoEntero pPosY);
public:
    CRobot();
   CRobot(TipoString pNombre, TipoCaracter pColor,
         TipoEntero pPosX, TipoEntero pPosY);
   virtual ~CRobot();
   TipoEntero    m_PosX;
   TipoEntero    m_PosY;
                                  … continua
```

```
    ....
  void setNombre(TipoString pNombre);
  TipoString getNombre();
  TipoEntero getPosX();
  TipoEntero getPosY();
  TipoCaracter getColor();
  TipoString getPosicion();
};
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Ejemplo de Agregación en C++ (BattleBots)

CBattleBox.h

```
#include "CRobot.h"
const TipoEntero ALTURA = 20;
const TipoEntero ANCHO = 20;
const TipoCaracter COLOR = '.';
class CBattleBox {
 private:
   vector<vector<TipoCaracter>> m_Plano;
   vector<CRobot*> m_Robots;
  void resizePlano(TipoEntero pAltura,
                   TipoEntero pAncho);
public:
   CBattleBox();
   CBattleBox(TipoEntero pAltura,
              TipoEntero pAncho);
   virtual ~CBattleBox();                    … continua
```

```
   …
   void adicionarRobot(CRobot* pRobot);
   CRobot* removerRobot(TipoString
pNombre);
   void imprimirRobots();
   TipoEntero getAltura();
   TipoEntero getAncho();
   TipoEntero getCantidadObjectos();
   void dibujarBattleBox();
   void actualizarBattleBox();
```

Uso de vector<...> para agregar robots

Ejemplo de Agregación en C++ (BattleBots)

CBattleBox.h

```
#include "CRobot.h"
const TipoEntero ALTURA = 20;
const TipoEntero ANCHO = 20;
const TipoCaracter COLOR = '.';
class CBattleBox {
 private:
   TipoEntero m_Altura;
   TipoEntero m_Ancho;
   TipoEntero n_CantRobots;
    TipoCaracter ** m_Plano;
    CRobot ** m_Robots = nullptr;
    void resizePlano(TipoEntero pAltura,
                     TipoEntero  pAncho)
public:
   CBattleBox();
   CBattleBox(TipoEntero pAltura, TipoEntero pAncho);
   virtual ~CBattleBox();
                                        … continua
```

```
...
void adicionarRobot(CRobot* pRobot);
CRobot* removerRobot(TipoString pNombre);
void imprimirRobots();
TipoEntero getAltura();
TipoEntero getAncho();
TipoEntero getCantidadObjectos();
void dibujarBattleBox();
void actualizarBattleBox();
```

Uso de PUNTEROS para agregar robots

Ejemplo de Agregación en C++ (BattleBots)

CBattleBox.cpp

```
.....

void CBattleBox::resizePlano(TipoEntero pAltura,
TipoEntero pAncho) {
    m_Plano.resize(pAltura);
   for ( auto & item : m_Plano )
       item.resize(pAncho);
}
CBattleBox::CBattleBox() {
        resizePlano(ALTURA, ANCHO);
}
CBattleBox::CBattleBox(TipoEntero pAltura,
TipoEntero pAncho) {
        resizePlano(pAltura, pAncho);
}
CBattleBox::~CBattleBox() {}
```

Llamada a función **PRIVADA**

Inicializando el Plano para agregar robots mediante VECTOR<>

Ejemplo de Agregación en C++ (BattleBots)

CBattleBox.cpp

```
....
void  CBattleBox::resizePlano(TipoEntero pAltura, TipoEntero pAncho){
  m_Plano = new TipoCaracter*[pAltura];
  for (size_t i = 0; i < pAltura; ++i)
    m_Plano[i] = new TipoCaracter[pAncho];
  }
CBattleBox::CBattleBox(): m_CantRobots{0},
 m_Altura{ALTURA}, m_Ancho{ANCHO}  {
  resizePlano(m_Altura, m_Ancho) ;
}
CBattleBox::CBattleBox(TipoEntero pAltura, TipoEntero pAncho) :
 m_Altura {pAltura}, m_Ancho {pAncho}, m_CantRobots{0} {
     resizePlano(m_Altura, m_Ancho) ;
}
CBattleBox::~CBattleBox() {
 for (size_t i = 0; i < m_Altura; ++i)
   delete[] m_Plano[i];
 delete[] m_Plano;
 m_Plano = nullptr;
}
```

Inicializando el Plano para agregar robots mediante PUNTEROS

Ejemplo de Agregación en C++ (BattleBots)

CBattleBox.cpp

```
....


void CBattleBox::adicionarRobot(CRobot* pRobot)
 {
   m_Robots.emplace_back(pRobot);
}
```

Agregando robots a la clase CBATTLEBOX mediante VECTOR<>

Ejemplo de Agregación en C++ (BattleBots)

CBattleBox.cpp

```
...


void CBattleBox::adicionarRobot(CRobot* pRobot) {
 CRobot** temp = new CRobot*[m_CantRobots + 1];
 for (size_t i  = 0; i < m_CantRobots; ++i)
   temp[i] = m_Robots[i];


 temp[m_CantRobots] = pRobot;


 delete [] m_Robots;
 m_Robots = temp;
 m_CantRobots++;
}
```

Agregando robots a la clase CBATTLEBOX mediante PUNTEROS

Ejemplo de Agregación en C++ (BattleBots)

CBattleBox.cpp

```
...
CRobot* CBattleBox::removerRobot(TipoString pNombre) {
  // Buscando objeto
  if (m_Robots.size() == 0)
    return nullptr;

  auto iter = find_if(begin(m_Robots), end(m_Robots),
                [pNombre](CRobot* obj){ return obj->getNombre() == pNombre; });
  if (iter == end(m_Robots))
    return nullptr;
  // Eliminando objeto
  m_Robots.erase(iter);
  return *iter;
}
```

Buscando el robot a eliminar

Eliminando robots de la clase CBATTLEBOX mediante VECTOR<>

Ejemplo de Agregación en C++ (BattleBots)

CBattleBox.cpp

```
CRobot* CBattleBox::removerRobot(TipoString
pNombre) {
 // Buscando objeto
 auto robot = buscarRobot(nombre);
 if (robot == nullptr) return nullptr;
 // Eliminando objeto
 CRobot** temp = new CRobot*[m_CantRobots - 1];
 int j = 0;
 for (size_t i  = 0; i < m_CantRobots; ++i)
   if (m_Robots[i] != robot) {
     temp[j] = m_Robots[i];
     j++;
   }
 m_Robots = temp;
 m_CantRobots--;
 return robot;
}
```

Buscando el robot a eliminar

```
CRobot* CBattleBox::buscarRobot(TipoString
pNombre) {
 for (size_t i  = 0; i < m_CantRobots; ++i) {
   if (pNombre == m_Robots[i]->getNombre()) {
     return m_Robots[i];
   } }
 return nullptr;
}
```

Eliminando robots de la clase CBATTLEBOX mediante PUNTEROS

Ejemplo de Agregación en C++ (BattleBots)

CBattleBox.cpp

```
void CBattleBox::actualizarBattleBox() {
  for (auto &row: m_Plano)
     for (auto &cell: row)
        cell = COLOR;

  for (auto& item: m_Robots)

m_Plano[item->getPosX()][item->getPosY()]
           = item->getColor();
}
```

```
void CBattleBox::actualizarBattleBox() {
for (size_t i = 0; i < m_Altura ; ++i) {
  for (size_t j = 0; j < m_Ancho; ++j) {
    m_Plano[i][j] = COLOR;
  }
}
for (size_t k = 0; k < m_CantRobots; ++k) {
m_Plano[m_Robots[k]->getPosX()][m_Robots[k]->getPosY()]
    = m_Robots[k]->getColor();
}
}
```

Actualizando BattleBox
con VECTOR y PUNTEROS

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

El programa lo pueden encontrar en repl.it

Usando vector : http://bit.ly/31ODQEP
Usando punteros: http://bit.ly/368ix4t

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

- ✔ ¿En qué consiste la relación entre clases?
- ✔ ¿En qué consiste la relación de Composición?
- ✔ ¿En qué consiste la relación de Asociación ?
- ✔ ¿En qué consiste la relación de Agregación?
- ✔ ¿Por qué usar Vector o Punteros para la Agregación?

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Resumen

En esta sesión aprendimos:

- Relaciones entre clases
  - Asociación
  - Composición
  - Agregación

¡Nos vemos en la siguiente clase!

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA