

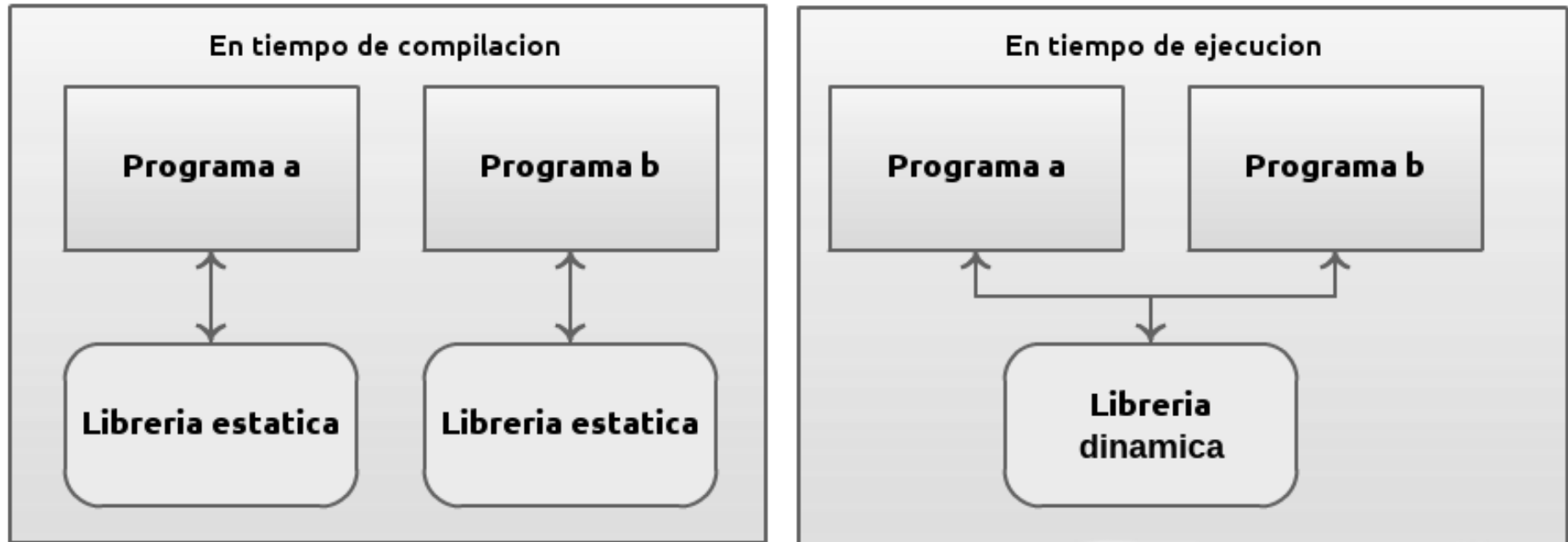


Creación y uso de librerías estáticas y dinámicas

Asignatura: PGE 2015

PGE: Librerías estáticas y dinámicas

Estática vs. dinámica



Estática

Dinámica

Windows

.a

.dll (Dynamic-Link Library)

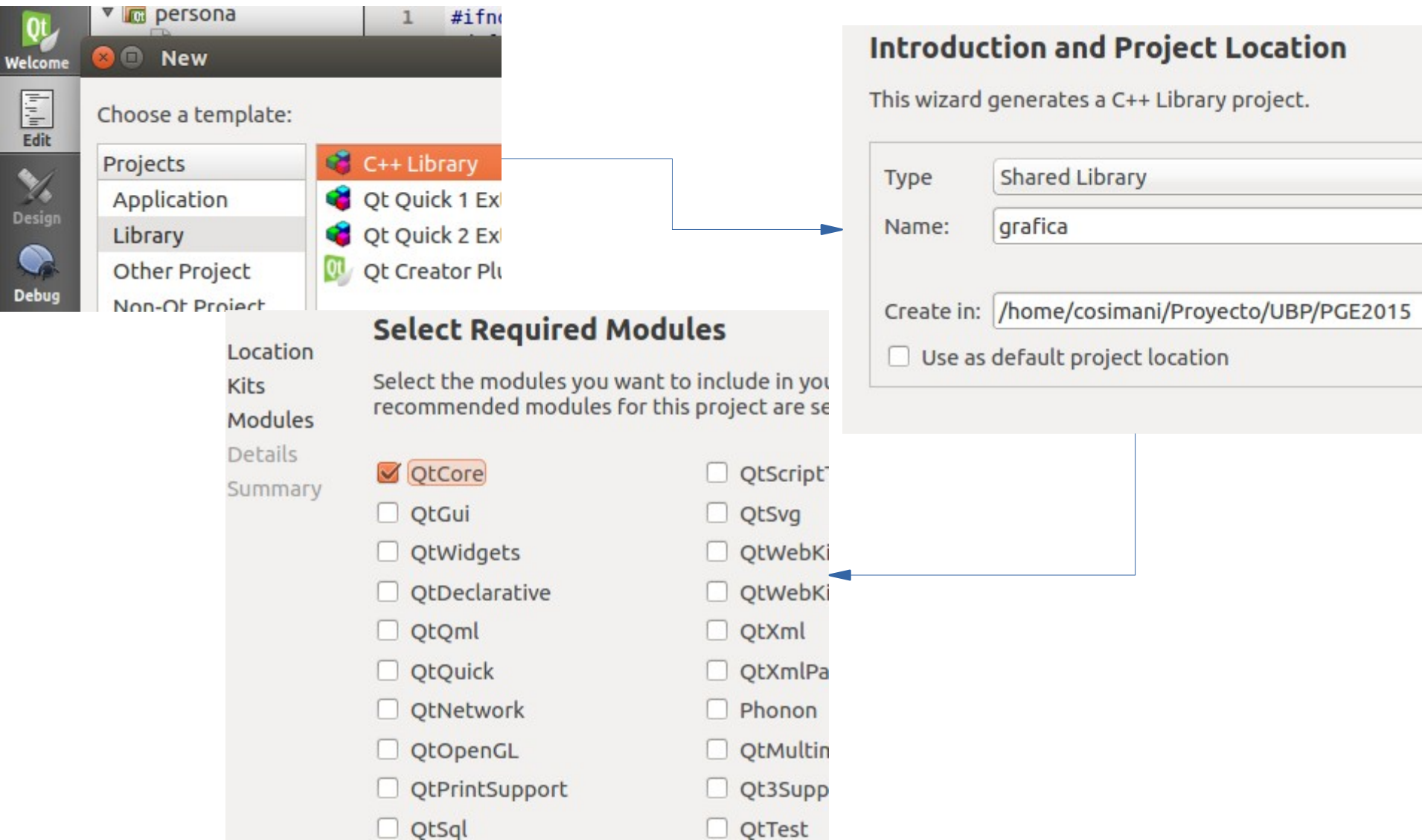
Linux

.a

.SO (Shared Object)

PGE: Librerías estáticas y dinámicas

Creación de librería dinámica con QtCreator



PGE: Librerías estáticas y dinámicas

Creación de librería dinámica con QtCreator

```
#ifndef PERSONA_H
#define PERSONA_H

#include "persona_global.h"

class PERSONASHARED_EXPORT Persona
{
public:
    Persona();

private:
    int edad;
};

#endif // PERSONA_H
```

```
QT      -= gui

TARGET = persona
TEMPLATE = lib

DEFINES += PERSONA_LIBRARY

SOURCES += persona.cpp

HEADERS += persona.h\
          persona_global.h
```

- Compilamos y vemos que se crea lo siguiente:
libpersona.a // Nos permitirá compilarlo en otro proyecto
persona.dll // Ese otro proyecto usará este dll
- Estos archivos los copiamos a una carpeta de uso común.
C:/proyectos/lib -> copiar los .a
C:/proyectos/dll -> copiar los .dll

***PGE:** Librerías estáticas y dinámicas*

Uso de librería dinámica con QtCreator

QT += core gui

INCLUDEPATH += includes

LIBS += -L"/home/cosimani/Proyecto/UBP/PGE2015/librerias/lib"

LIBS += -lpersona

SOURCES += main.cpp\
principal.cpp

HEADERS += principal.h

o bien **INCLUDEPATH** += C:/proyectos/uso/includes
INCLUDEPATH += includes

LIBS += C:/proyectos/lib/libpersona.a

o bien indicando el directorio y archivos por separado:

LIBS += -L" C:/proyectos/lib"

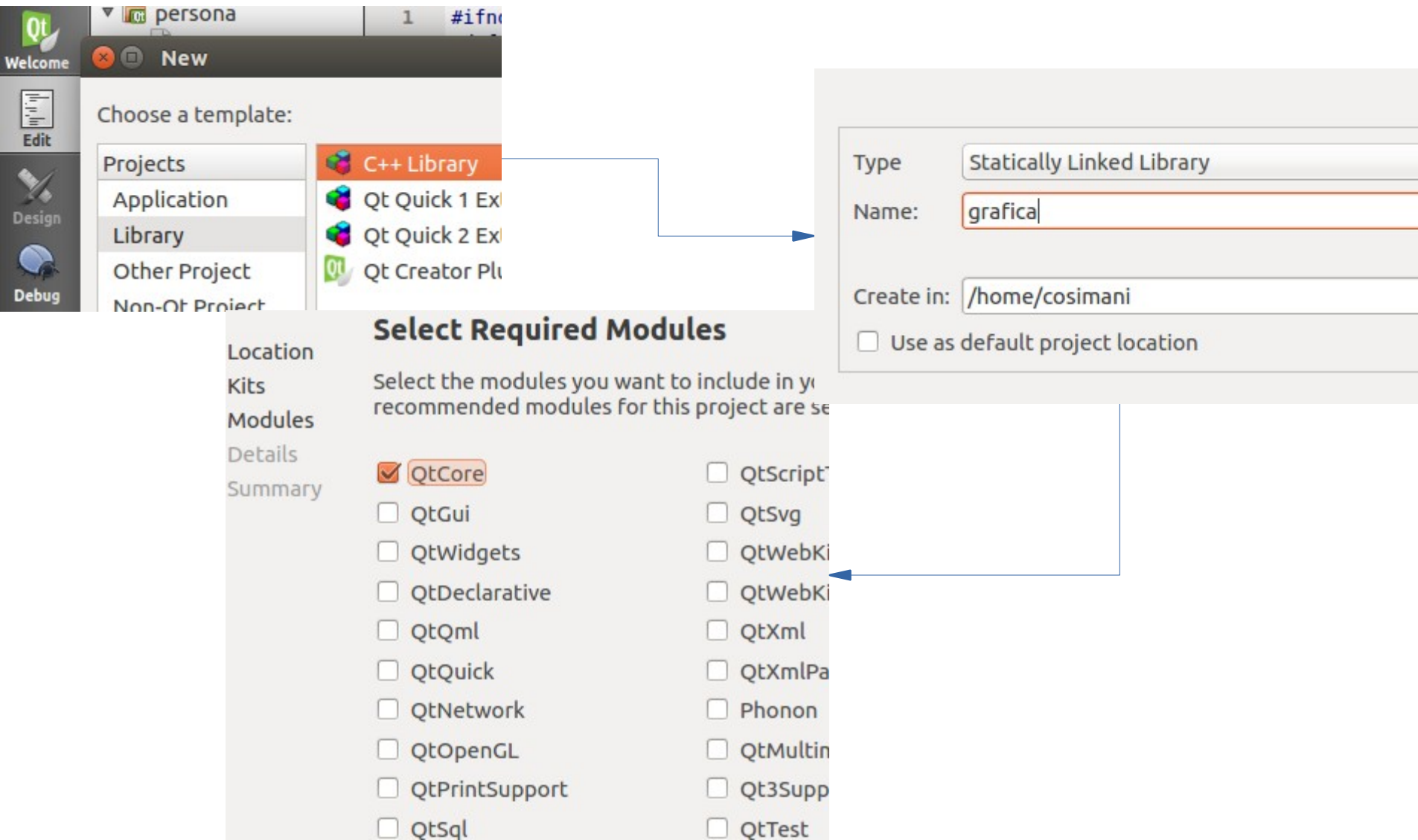
LIBS += -lpersona

- Para especificar el directorio usar -L
- Para el archivo -l
- Notar que aquí no se indica el sufijo lib ni la extensión .a

- Además de esto debemos tener acceso a persona.dll al ejecutar el programa.
- Para esto hay 3 posibilidades:
 - Agregar al PATH la ruta C:/proyectos/dll
 - Copiar el dll dentro de la carpeta donde se encuentra el ejecutable
 - Agregar el PATH sólo desde Qt - Projects - Build Environment

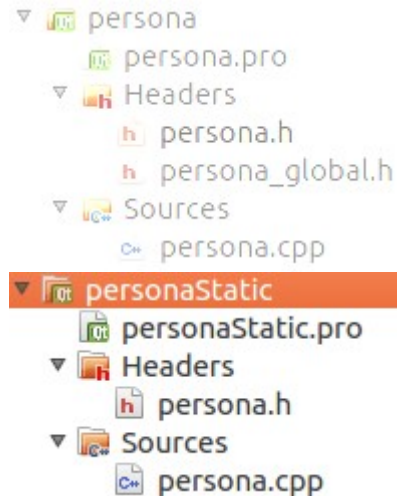
PGE: Librerías estáticas y dinámicas

Creación de librería estática con QtCreator



PGE: Librerías estáticas y dinámicas

Creación de librería estática con QtCreator



```
#ifndef PERSONA_H
#define PERSONA_H

class Persona {
public:
    Persona();

private:
    int edad;
};

#endif // PERSONA_H

QT     -= gui

TARGET = persona
TEMPLATE = lib

DEFINES += PERSONA_LIBRARY

SOURCES += persona.cpp

HEADERS += persona.h\
          persona_global.h
```

```
#ifndef PERSONA_H
#define PERSONA_H

#include "persona_global.h"

class PERSONASHARED_EXPORT Persona
{
public:
    Persona();

private:
    int edad;
};

#endif // PERSONA_H

QT     -= gui

TARGET = persona
TEMPLATE = lib
CONFIG += staticlib

SOURCES += persona.cpp

HEADERS += persona.h
```

- Compilamos y vemos que se crea lo siguiente:
libpersona.a // Nos permitirá compilarlo en otro proyecto
- Copiar el .a en una carpeta de uso común.
C:/proyectos/lib -> copiar los .a

PGE: Librerías estáticas y dinámicas

Uso de librería estática con QtCreator

```
QT      += core gui

INCLUDEPATH += includes

LIBS += -L"/home/cosimani/Proyecto/UBP/PGE2015/librerias/lib"
LIBS += -lpersona

SOURCES += main.cpp\
          principal.cpp
HEADERS += principal.h
```

o bien

```
INCLUDEPATH += C:/proyectos/uso/includes
INCLUDEPATH += includes
```

```
LIBS += C:/proyectos/lib/libpersona.a
```

o bien indicando el directorio y archivos por separado:

```
LIBS += -L" C:/proyectos/lib"
LIBS += -lpersona
```

- Para especificar el directorio usar -L
- Para el archivo -l
- Notar que aquí no se indica el sufijo lib ni la extensión .a

- Aquí no necesitamos de ningún dll porque el .a se incrusta en el .exe
- Este .exe pesará más que el que usa los dll

PGE: Librerías estáticas y dinámicas

Linux vs. Windows

Buscar dependencias de librerías

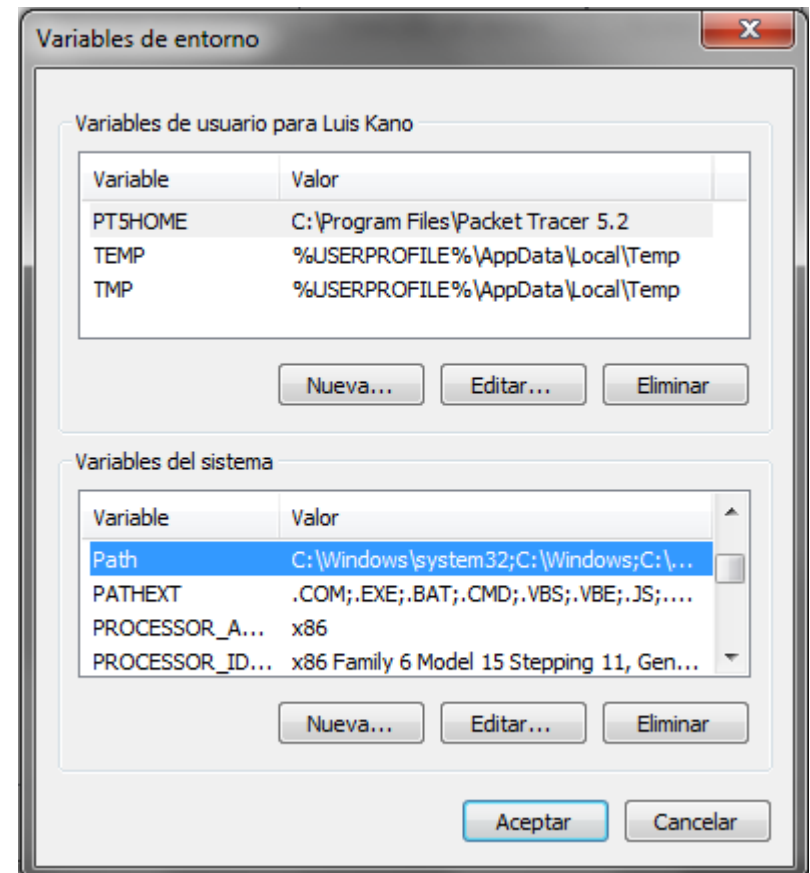
Linux → Comando `ldd` (List Dynamic Dependencies)
Windows → Aplicación `dependencyWalker`

Linux

```
export LD_LIBRARY_PATH=./  
export LD_LIBRARY_PATH=/home/dll
```

```
echo $LD_LIBRARY_PATH
```

Windows



Ejercicio

- *Construir una librería dinámica*
- *Debe contener las siguientes clases:*
 - *Clase Persona bien completa.*
 - *Clase Login*
- *Dejar disponible los siguientes archivos:*
 - *libpropia.a*
 - *propia.dll*
 - *login.h persona.h propia_global.h*
- *Desarrollar un proyecto que utilice la clase Login que se encuentra en esta librería dinámica*