

Clases/Métodos de instancia/métodos de clase

Ejecución del programa:

| main |

main := AAEstudioCinematografico new.

main mostrarMenu

AAEstudioCinematografico

instanceVariableNames: 'listadoPeliculas listadoPersonal'

classVariableNames: "

buscarPelicula

```
| pelicula objeto |
listadoPeliculas isEmpty
  ifTrue: [MessageBox notify: 'No hay peliculas cargadas']
  ifFalse:
    [[objeto isNil] whileTrue:
      [pelicula := Prompter prompt: 'Ingrese el
nombre de la pelicula'.
      objeto := listadoPeliculas detect: [:i | pelicula =
i obtenerTituloPelicula]
      ifNone:
        [objeto := nil.
        MessageBox notify:
'No hay ninguna pelicula con ese titulo']]].
^objeto
```

buscarPersonal

```
| personal objeto |
listadoPersonal isEmpty
  ifTrue: [MessageBox notify: 'No hay ningun empleado cargado']
  ifFalse:
    [[objeto isNil] whileTrue:
      [personal := Prompter prompt: 'Ingrese el
nombre del personal'.
      objeto := listadoPersonal detect: [:i | personal
= i obtenerNombrePersonal]
      ifNone:
        [objeto := nil.
        MessageBox notify:
'El personal no existe']]].
^objeto
```

busquedaEspecialidad

```

| colec especialidad |
especialidad := Prompter
                                prompt: 'Ingrese la especialidad a buscar.
S(Sonidista) | I(Iluminador) | C(Camarografo)'.
                                colec := listadoPersonal select: [:i | i obtenerEspecialidad = especialidad].
                                colec isEmpty
                                ifTrue: [MessageBox notify: 'No hay ningun empleado con esa
especialidad']
                                ifFalse: [self listarPersonal: colec]

```

cargarEstudio

```

listadoPeliculas := OrderedCollection new.
listadoPersonal := OrderedCollection new.
AAPelicula inicializarCodigo.
MessageBox notify: 'Estudio Cinematografico CINEMA'.
PersonalPermanente cargarSueldoBasico.
self cargarPersonalEstudio

```

cargarPersonalEstudio

```

| nuevo personal |
MessageBox notify: 'Cargar los empleados del estudio'.
nuevo := true.
[nuevo] whileTrue:
    [personal := PersonalPermanente new.
    personal cargarStaff.
    listadoPersonal add: personal.
    nuevo := MessageBox confirm: '¿Desea agregar otro
empleado?']

```

listarPeliculas: lista

```

Transcript
clear;

```

```
cr;  
show: 'COD';  
tab;  
tab;  
show: 'TITULO';  
tab;  
tab;  
show: 'PRES ASIG';  
tab;  
tab;  
show: 'PRES REM';  
cr.
```

lista do: [:i | i mostrarPelicula]

listarPersonal: lista

Transcript

```
clear;  
cr;  
show: 'NOMBRE';  
tab;  
tab;  
show: 'APELLIDO';  
tab;  
tab;  
show: 'DOC';  
tab;  
tab;  
show: 'PLUS';  
tab;  
tab;
```

```
show: 'ESP';  
cr.  
lista do: [:i | i mostrarPersonal]
```

mostrarMenu

```
| opcion |  
self cargarEstudio.  
opcion := nil.  
[opcion = '0'] whileFalse:  
    [MessageBox  
        notify: '1- Cargar una pelicula. 2- Cargar un  
actor/director. 3- Calcular plus de un empleado. 4-Ver listado de peliculas. 5-  
Ver listado de empleados. 0- Salir'.  
opcion := Prompter prompt: 'Elija una opcion'.  
opcion = '1' ifTrue: [self opcionCargarPelicula].  
opcion = '2' ifTrue: [self opcionCargarStaff].  
opcion = '3' ifTrue: [self opcionCalcularPlus].  
opcion = '4' ifTrue: [self opcionListaPeliculas].  
opcion = '5' ifTrue: [self opcionListaPersonal]]
```

opcionCalcularPlus

```
| pelicula personal pago |  
pelicula := self buscarPelicula.  
pelicula isNil  
    ifFalse:  
        [personal := self buscarPersonal.  
personal isNil  
        ifFalse:  
            [pago := pelicula calcularPlus: personal.  
            MessageBox  
                notify: 'En la pelicula ' , pelicula  
obtenerTituloPelicula , ' el empleado ']
```

```
obtenerNombrePersonal , ' obtendra un plus de $'
, personal
, pago printString]]
```

opcionCargarActor

```
| actor pelicula |
pelicula := self buscarPelicula.
pelicula isNil
    ifFalse:
        [actor := Actor new.
        actor cargarStaff.
        pelicula agregarActor: actor]
```

opcionCargarDirector

```
| director pelicula pago |
pelicula := self buscarPelicula.
pelicula isNil
    ifFalse:
        [director := EquipoDireccion new.
        director cargarStaff.
        pago := pelicula obtenerPresAsig * director
obtenerPorcentaje / 100.
        director agregarPago: pago.
        pelicula agregarDirector: director]
```

opcionCargarPelicula

```
| pelicula nuevo personal |
pelicula := AAPelicula new.
pelicula cargarPelicula.
    MessageBox notify: 'Ingrese el personal del estudio que trabajara en la
pelicula'.
    nuevo := true.
```

```
[nuevo] whileTrue:  
    [personal := self buscarPersonal.  
    personal isNil ifFalse: [pelicula agregarPersonal: personal].  
    nuevo := MessageBox confirm: '¿Desea agregar otro  
personal?'].  
    listadoPelículas add: pelicula
```

opcionCargarStaff

```
| opcion |  
opcion := nil.  
[opcion = '0'] whileFalse:  
    [MessageBox notify: '1- Cargar un actor. 2- Cargar un  
director. 0- Salir'.  
    opcion := Prompter prompt: 'Elija una opcion'.  
    opcion = '1' ifTrue: [self opcionCargarActor].  
    opcion = '2' ifTrue: [self opcionCargarDirector]]
```

opcionListaPelículas

```
self listarPelículas: listadoPelículas
```


opcionListaPersonal

self busquedaEspecialidad

AAPelicula

instanceVariableNames: 'codigo titulo presAsig presRem listadoStaff'

classVariableNames: 'Codigo'

agregarActor: actor

presRem := presRem - actor obtenerCachet.

listadoStaff add: actor

agregarDirector: director

presRem := presRem - director obtenerPagoDirector.

listadoStaff add: director

agregarPersonal: personal

presRem := presRem - PersonalPermanente obtenerSueldoBasico.

listadoStaff add: personal

calcularPlus: personal

| pago |

pago := presRem * personal obtenerPlus / 100.

^pago

cargarPelicula

codigo := Codigo.

AAPelicula actualizarCodigo.

titulo := Prompter prompt: 'Ingrese el titulo de la pelicula ' , codigo
printString.

presAsig := (Prompter prompt: 'Ingrese el presupuesto de la pelicula ' ,
titulo) asNumber.

presRem := presAsig.

listadoStaff := OrderedCollection new

mostrarPelicula

Transcript

cr;

show: codigo printString;

```
    tab;  
    tab;  
    show: titulo asUppercase;  
    tab;  
    tab;  
    show: presAsig printString;  
    tab;  
    tab;  
    show: presRem printString;  
    cr
```

obtenerPresAsig

^presAsig

obtenerPresRem

^presRem

obtenerTituloPelicula

^titulo

actualizarCodigo

Codigo := Codigo + 1

inicializarCodigo

Codigo := 1

AAStaff

instanceVariableNames: 'nombre apellido documento'

classVariableNames: "

cargarStaff

nombre := Prompter prompt: 'Ingrese el nombre de la persona'.

apellido := Prompter prompt: 'Ingrese el apellido de la persona'.

documento := (Prompter prompt: 'Ingrese el documento de identidad')
asNumber

AAStaff subclass: #Actor

instanceVariableNames: 'nacionalidad cachet'

classVariableNames: ''

cargarStaff

super cargarStaff.

nacionalidad := Prompter prompt: 'Ingrese la nacionalidad del actor'.

cachet := (Prompter prompt: 'Ingrese el cachet a cobrar por el actor')
asNumber

obtenerCachet

^cachet

AAStaff subclass: #EquipoDireccion

instanceVariableNames: 'nacionalidad porcentaje pagoDirector'

classVariableNames: ''

agregarPago: pago

pagoDirector := pago

cargarStaff

super cargarStaff.

nacionalidad := Prompter prompt: 'Ingrese la nacionalidad del director'.

porcentaje := (Prompter prompt: 'Ingrese el porcentaje a cobrar por el director') asNumber asFloat

obtenerPagoDirector

^pagoDirector

obtenerPorcentaje

^porcentaje

AAStaff subclass: #PersonalPermanente

instanceVariableNames: 'plus especialidad'

classVariableNames: 'Sueldo'

cargarStaff

super cargarStaff.

plus := (Prompter prompt: 'Ingrese el porcentaje a cobrar por el empleado') asNumber asFloat.

especialidad := (Prompter

prompt: 'Ingrese la especialidad del empleado.

S(Sonidista) | I(Iluminador) | C(Camarografo)')

asUppercase

mostrarPersonal

Transcript

cr;

show: nombre asUppercase;

tab;

tab;

show: apellido asUppercase;

tab;

tab;

show: documento printString;

tab;

tab;

show: plus printString;

tab;

tab;

show: especialidad asUppercase;

cr

obtenerEspecialidad

^especialidad

obtenerNombrePersonal

^nombre

obtenerPlus

^plus

cargarSueldoBasico

Sueldo := (Prompter prompt: 'Ingrese el sueldo basico de los empleados del estudio') asNumber

obtenerSueldoBasico

^Sueldo