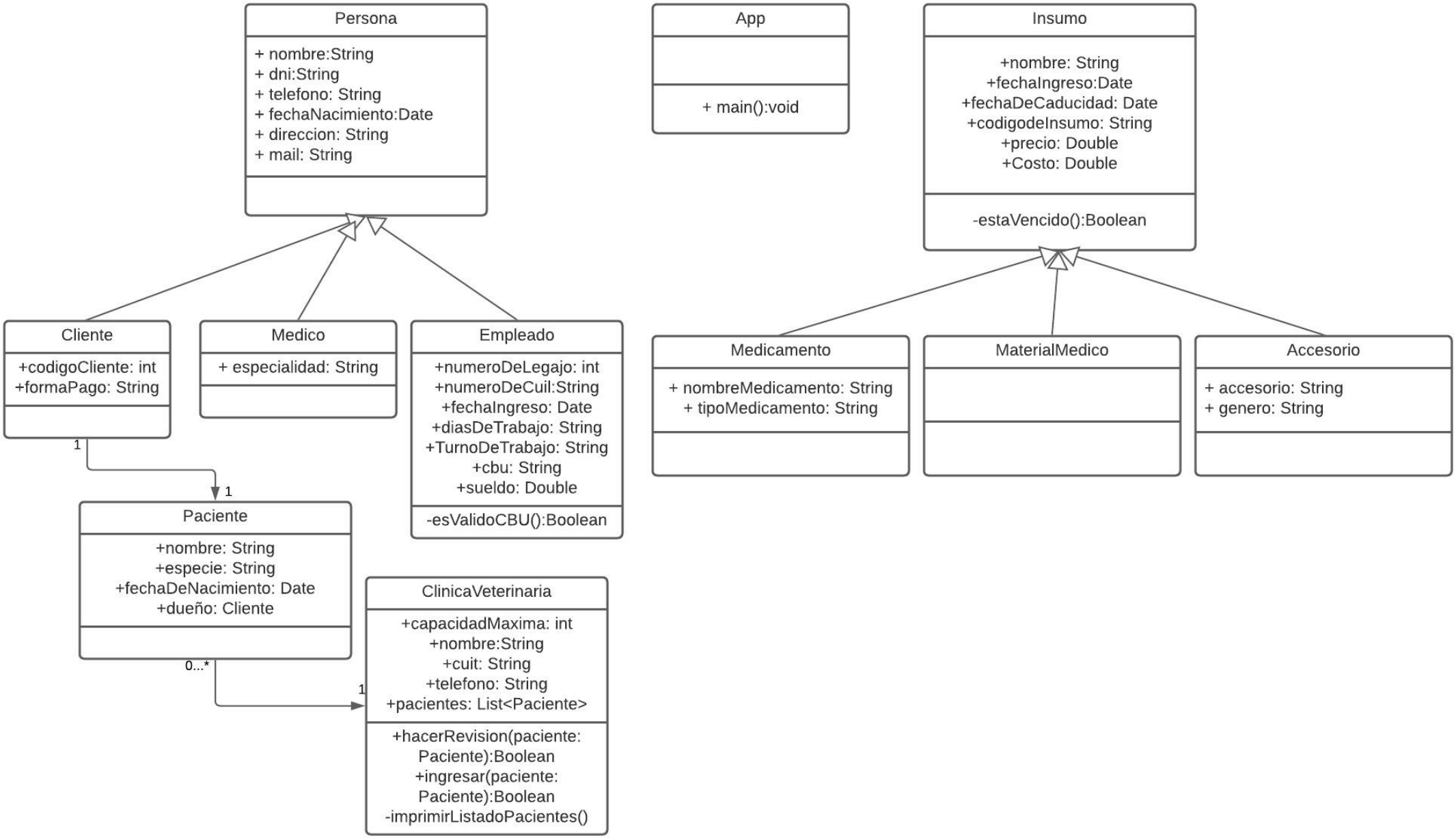
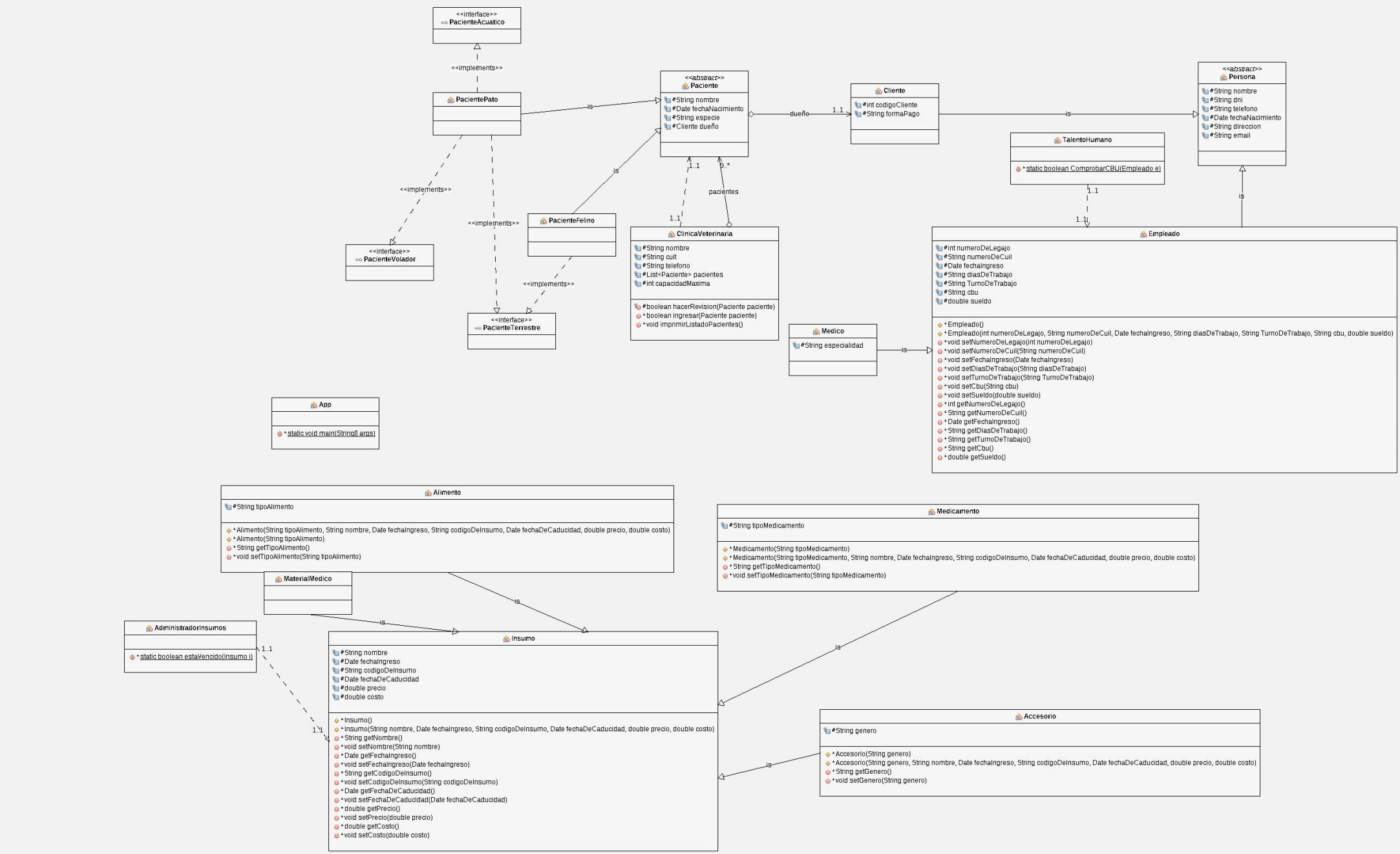
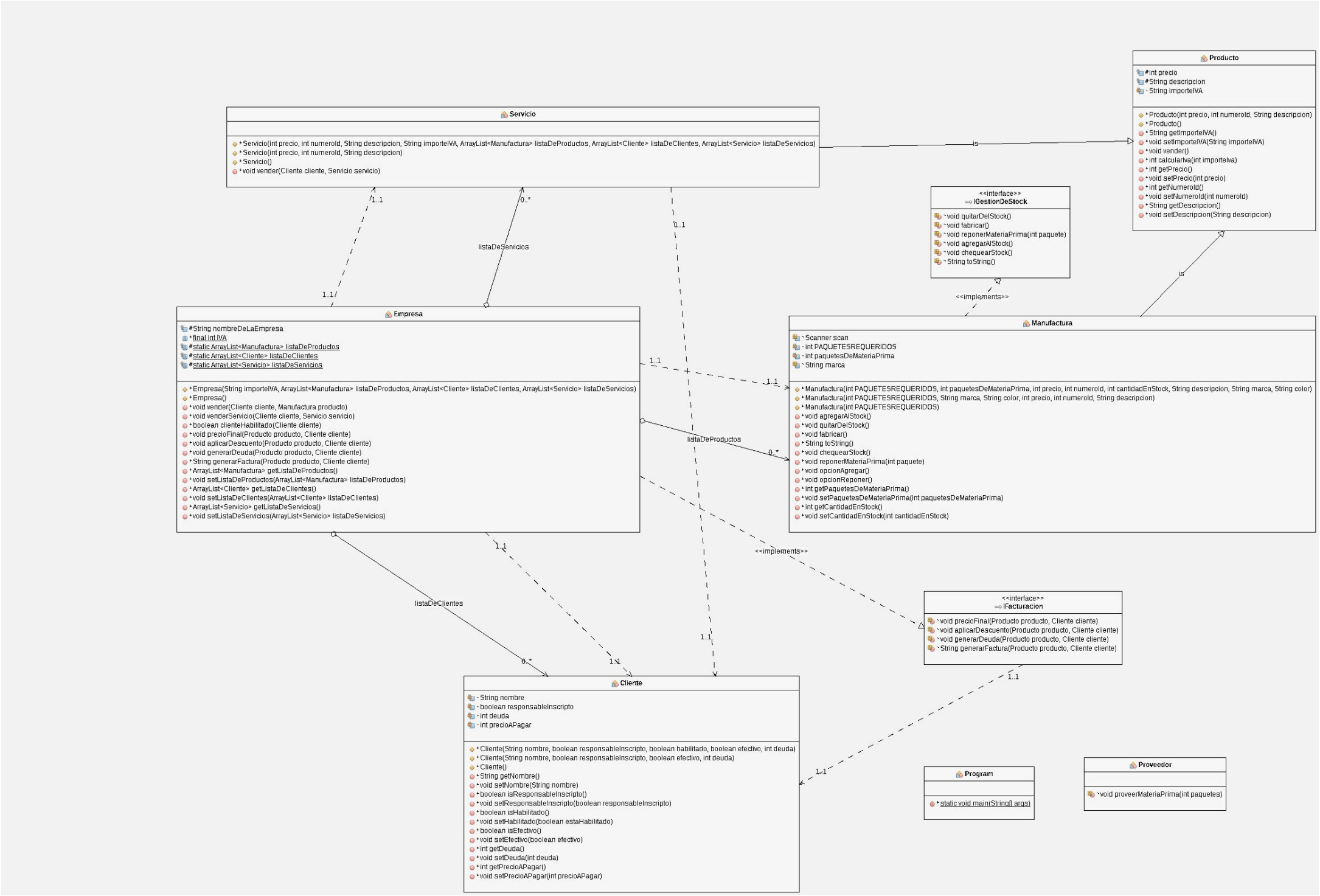
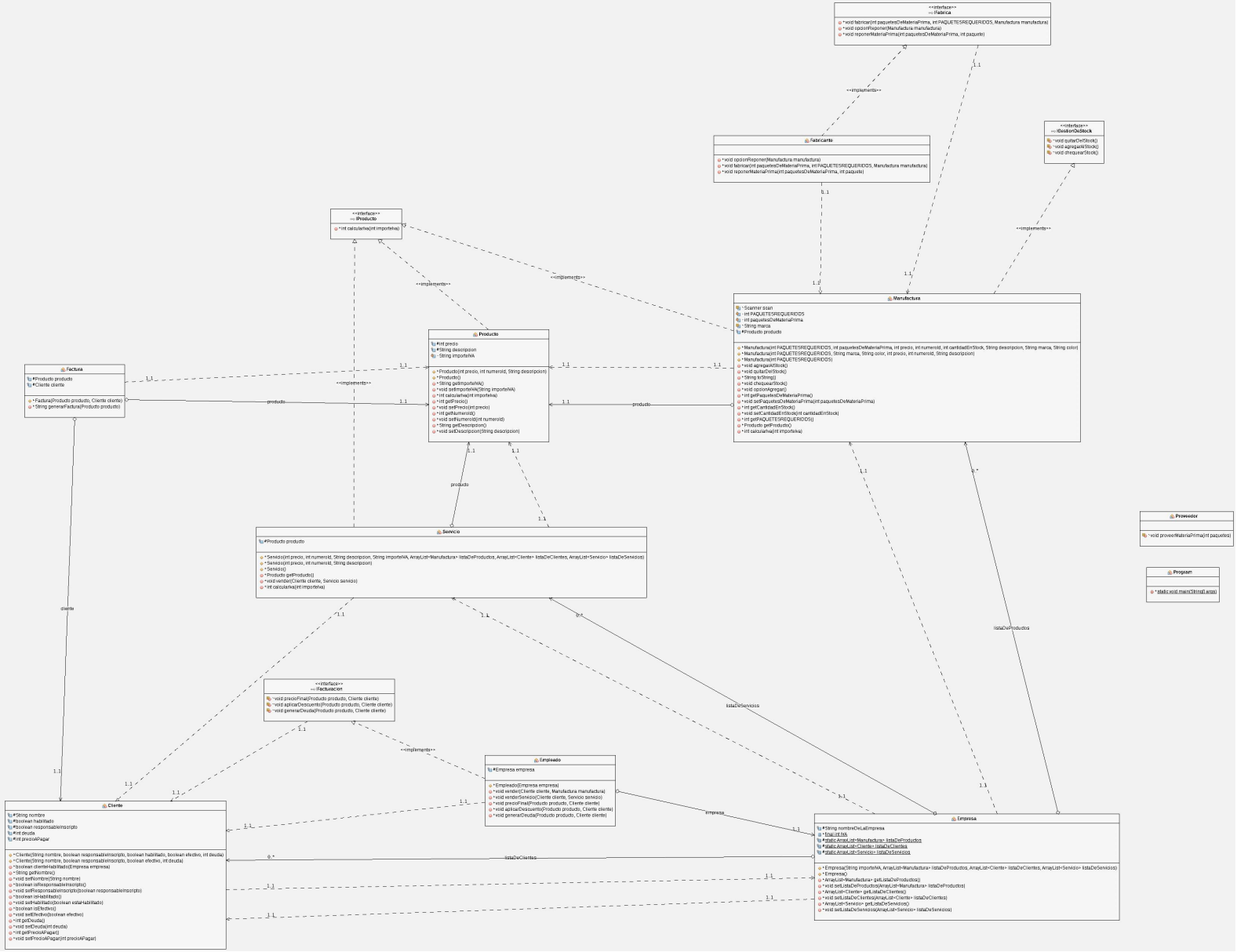
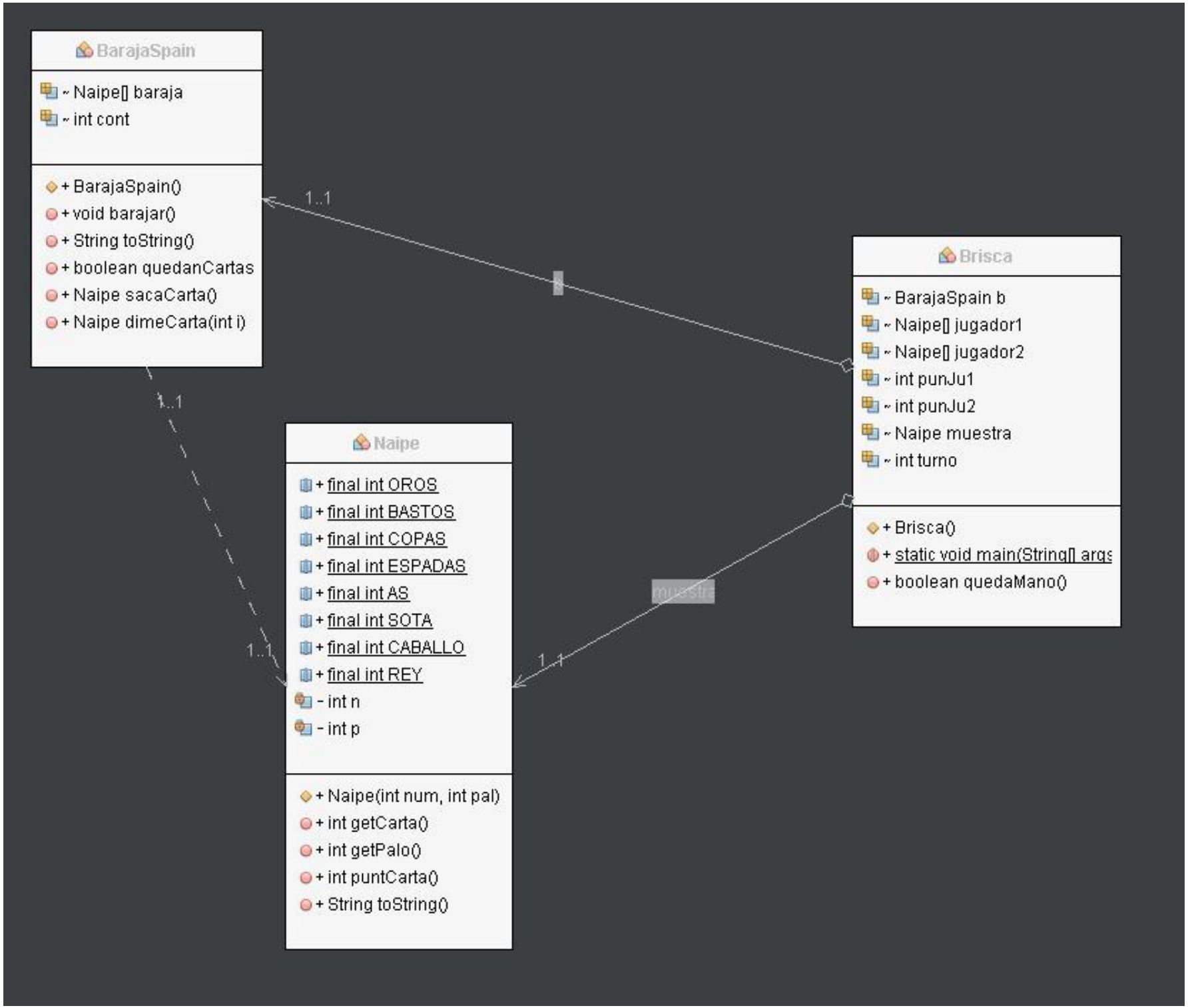
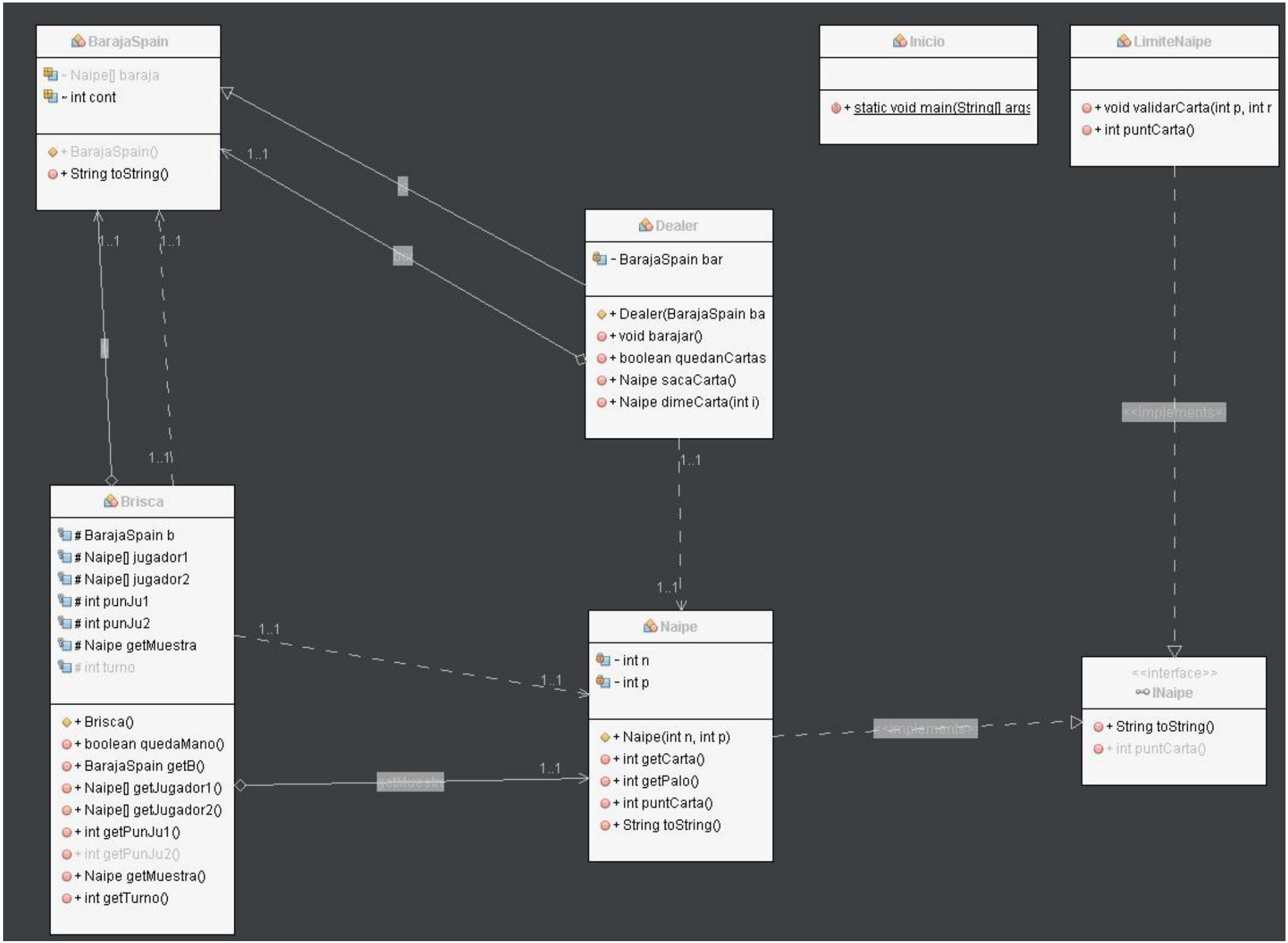

Persona
+ nombre:String
+ dni:String
+ telefono: String
+ fechaNacimiento:Date
+ direccion: String
+ mail: String
App
+ main():void
Insumo
+nombre: String
+fechaIngreso:Date
+fechaDeCaducidad: Date
+codigodeInsumo: String
+precio: Double
+Costo: Double
-estaVencido():Boolean
Cliente
+codigoCliente: int
+formaPago: String
Medico
+ especialidad: String
Empleado
+numeroDeLegajo: int
+numeroDeCuil:String
+fechaIngreso: Date
+diasDeTrabajo: String
+TurnoDeTrabajo: String
+cbu: String
+sueldo: Double
-esValidoCBU():Boolean
Medicamento
+ nombreMedicamento: String
+ tipoMedicamento: String
MaterialMedico
Accesorio
+ accesorio: String
+ genero: String
1
1
Paciente
+nombre: String
+especie: String
+fechaDeNacimiento: Date
+dueño: Cliente
0...*
1
ClinicaVeterinaria
+capacidadMaxima: int
+nombre:String
+cuit: String
+telefono: String
+pacientes: List<Paciente>
+hacerRevision(paciente: Paciente):Boolean
+ingresar(paciente: Paciente):Boolean
-imprimirListadoPacientes()


<<interface>>
PacienteAcuatico
<<implements>>
PacientePato
is
<<abstract>>
Paciente
#String nombre
#Date fechaNacimiento
#String especie
#Cliente dueño
dueño
1..1
Cliente
#int codigoCliente
#String formaPago
is
<<abstract>>
Persona
#String nombre
#String dni
#String telefono
#Date fechaNacimiento
#String direccion
#String email
TalentoHumano
*static boolean ComprobarCBU(Empleado e)
<<implements>>
<<interface>>
PacienteVolador
<<implements>>
<<interface>>
PacienteTerrestre
is
PacienteFelino
<<implements>>
1..1
0..*
pacientes
ClinicaVeterinaria
#String nombre
#String cuit
#String telefono
#List<Paciente> pacientes
#int capacidadMaxima
#boolean hacerRevision(Paciente paciente)
*boolean ingresar(Paciente paciente)
*void imprimirListadoPacientes()
1..1
1..1
is
Empleado
#int numeroDeLegajo
#String numeroDeCuil
#Date fechaIngreso
#String diasDeTrabajo
#String TurnoDeTrabajo
#String cbu
#double sueldo
*Empleado()
*Empleado(int numeroDeLegajo, String numeroDeCuil, Date fechaIngreso, String diasDeTrabajo, String TurnoDeTrabajo, String cbu, double sueldo)
*void setNumeroDeLegajo(int numeroDeLegajo)
*void setNumeroDeCuil(String numeroDeCuil)
*void setFechaIngreso(Date fechaIngreso)
*void setDiasDeTrabajo(String diasDeTrabajo)
*void setTurnoDeTrabajo(String TurnoDeTrabajo)
*void setCbu(String cbu)
*void setSueldo(double sueldo)
*int getNumeroDeLegajo()
*String getNumeroDeCuil()
*Date getFechaIngreso()
*String getDiasDeTrabajo()
*String getTurnoDeTrabajo()
*String getCbu()
*double getSueldo()
Medico
#String especialidad
is
App
*static void main(String[] args)
Alimento
#String tipoAlimento
*Alimento(String tipoAlimento, String nombre, Date fechaIngreso, String codigoDeInsumo, Date fechaDeCaducidad, double precio, double costo)
*Alimento(String tipoAlimento)
*String getTipoAlimento()
*void setTipoAlimento(String tipoAlimento)
Medicamento
#String tipoMedicamento
*Medicamento(String tipoMedicamento)
*Medicamento(String tipoMedicamento, String nombre, Date fechaIngreso, String codigoDeInsumo, Date fechaDeCaducidad, double precio, double costo)
*String getTipoMedicamento()
*void setTipoMedicamento(String tipoMedicamento)
MaterialMedico
is
is
is
AdministradorInsumos
*static boolean estaVencido(Insumo i)
1..1
1..1
Insumo
#String nombre
#Date fechaIngreso
#String codigoDeInsumo
#Date fechaDeCaducidad
#double precio
#double costo
*Insumo()
*Insumo(String nombre, Date fechaIngreso, String codigoDeInsumo, Date fechaDeCaducidad, double precio, double costo)
*String getNombre()
*void setNombre(String nombre)
*Date getFechaIngreso()
*void setFechaIngreso(Date fechaIngreso)
*String getCodigoDeInsumo()
*void setCodigoDeInsumo(String codigoDeInsumo)
*Date getFechaDeCaducidad()
*void setFechaDeCaducidad(Date fechaDeCaducidad)
*double getPrecio()
*void setPrecio(double precio)
*double getCosto()
*void setCosto(double costo)
is
Accesorio
#String genero
*Accesorio(String genero)
*Accesorio(String genero, String nombre, Date fechaIngreso, String codigoDeInsumo, Date fechaDeCaducidad, double precio, double costo)
*String getGenero()
*void setGenero(String genero)
is


Servicio
+Servicio(int precio, int numerold, String descripcion, String importeIVA, ArrayList<Manufactura> listaDeProductos, ArrayList<Cliente> listaDeClientes, ArrayList<Servicio> listaDeServicios)
+Servicio(int precio, int numerold, String descripcion)
+Servicio()
+void vender(Cliente cliente, Servicio servicio)
Producto
#int precio
#String descripcion
- String importeIVA
+Producto(int precio, int numerold, String descripcion)
+Producto()
+String getImporteIVA()
+void setImporteIVA(String importeIVA)
+void vender()
+int calcularIva(int importeIva)
+int getPrecio()
+void setPrecio(int precio)
+int getNumerold()
+void setNumerold(int numerold)
+String getDescripcion()
+void setDescripcion(String descripcion)
is
<<interface>>
IGestionDeStock
~void quitarDelStock()
~void fabricar()
~void reponerMateriaPrima(int paquete)
~void agregarAlStock()
~void chequearStock()
~String toString()
<<implements>>
is
listaDeServicios
1..1
0..*
1..1
1..1
Empresa
#String nombreDeLaEmpresa
+final int IVA
#static ArrayList<Manufactura> listaDeProductos
#static ArrayList<Cliente> listaDeClientes
#static ArrayList<Servicio> listaDeServicios
+Empresa(String importeIVA, ArrayList<Manufactura> listaDeProductos, ArrayList<Cliente> listaDeClientes, ArrayList<Servicio> listaDeServicios)
+Empresa()
+void vender(Cliente cliente, Manufactura producto)
+void venderServicio(Cliente cliente, Servicio servicio)
+boolean clienteHabilitado(Cliente cliente)
+void precioFinal(Producto producto, Cliente cliente)
+void aplicarDescuento(Producto producto, Cliente cliente)
+void generarDeuda(Producto producto, Cliente cliente)
+String generarFactura(Producto producto, Cliente cliente)
+ArrayList<Manufactura> getListaDeProductos()
+void setListaDeProductos(ArrayList<Manufactura> listaDeProductos)
+ArrayList<Cliente> getListaDeClientes()
+void setListaDeClientes(ArrayList<Cliente> listaDeClientes)
+ArrayList<Servicio> getListaDeServicios()
+void setListaDeServicios(ArrayList<Servicio> listaDeServicios)
1..1
1..1
listaDeProductos
0..*
Manufactura
~Scanner scan
- int PAQUETESREQUERIDOS
- int paquetesDeMateriaPrima
~String marca
+Manufactura(int PAQUETESREQUERIDOS, int paquetesDeMateriaPrima, int precio, int numerold, int cantidadEnStock, String descripcion, String marca, String color)
+Manufactura(int PAQUETESREQUERIDOS, String marca, String color, int precio, int numerold, String descripcion)
+Manufactura(int PAQUETESREQUERIDOS)
+void agregarAlStock()
+void quitarDelStock()
+void fabricar()
+String toString()
+void chequearStock()
+void reponerMateriaPrima(int paquete)
+void opcionAgregar()
+void opcionReponer()
+int getPaquetesDeMateriaPrima()
+void setPaquetesDeMateriaPrima(int paquetesDeMateriaPrima)
+int getCantidadEnStock()
+void setCantidadEnStock(int cantidadEnStock)
1..1
<<implements>>
listaDeClientes
0..*
1..1
1..1
<<interface>>
IFacturacion
~void precioFinal(Producto producto, Cliente cliente)
~void aplicarDescuento(Producto producto, Cliente cliente)
~void generarDeuda(Producto producto, Cliente cliente)
~String generarFactura(Producto producto, Cliente cliente)
1..1
1..1
Cliente
- String nombre
- boolean responsableInscripto
- int deuda
- int precioAPagar
+Cliente(String nombre, boolean responsableInscripto, boolean habilitado, boolean efectivo, int deuda)
+Cliente(String nombre, boolean responsableInscripto, boolean efectivo, int deuda)
+Cliente()
+String getNombre()
+void setNombre(String nombre)
+boolean isResponsableInscripto()
+void setResponsableInscripto(boolean responsableInscripto)
+boolean isHabilitado()
+void setHabilitado(boolean estaHabilitado)
+boolean isEfectivo()
+void setEfectivo(boolean efectivo)
+int getDeuda()
+void setDeuda(int deuda)
+int getPrecioAPagar()
+void setPrecioAPagar(int precioAPagar)
Program
+static void main(String[] args)
Proveedor
~void proveerMateriaPrima(int paquetes)


<<interface>>
Fabrica
+void fabricar(int paquetesDeMateriaPrima, int PAQUETESREQUERIDOS, Manufactura manufactura)
+void opcionReponer(Manufactura manufactura)
+void reponerMateriaPrima(int paquetesDeMateriaPrima, int paquete)
<<implementa>>
Fabricante
+void opcionReponer(Manufactura manufactura)
+void fabricar(int paquetesDeMateriaPrima, int PAQUETESREQUERIDOS, Manufactura manufactura)
+void reponerMateriaPrima(int paquetesDeMateriaPrima, int paquete)
<<interface>>
IGestionDeStock
-void quitarDeStock()
-void agregarAlStock()
-void chequearStock()
<<implementa>>
<<interface>>
IProducto
+int calcularIva(int importeIva)
<<implementa>>
<<implementa>>
Manufactura
-Scanner scan
-int PAQUETESREQUERIDOS
-int paquetesDeMateriaPrima
-String marca
#Producto producto
+Manufactura(int PAQUETESREQUERIDOS, int paquetesDeMateriaPrima, int precio, int numeroId, int cantidadEnStock, String descripcion, String marca, String color)
+Manufactura(int PAQUETESREQUERIDOS, String marca, String color, int precio, int numeroId, String descripcion)
+Manufactura(int PAQUETESREQUERIDOS)
+void agregarAlStock()
+void quitarDeStock()
+String toString()
+void chequearStock()
+void opcionAgregar()
+int getPaquetesDeMateriaPrima()
+void setPaquetesDeMateriaPrima(int paquetesDeMateriaPrima)
+int getCantidadEnStock()
+void setCantidadEnStock(int cantidadEnStock)
+int getPAQUETESREQUERIDOS()
+Producto getProducto()
+int calcularIva(int importeIva)
Producto
#int precio
#String descripcion
-String importeIVA
+Producto(int precio, int numeroId, String descripcion)
+Producto()
+String getImporteIVA()
+void setImporteIVA(String importeIVA)
+int calcularIva(int importeIva)
+int getPrecio()
+void setPrecio(int precio)
+int getNumeroId()
+void setNumeroId(int numeroId)
+String getDescripcion()
+void setDescripcion(String descripcion)
Factura
#Producto producto
#Cliente cliente
+Factura(Producto producto, Cliente cliente)
+String generarFactura(Producto producto)
<<implementa>>
producto
producto
producto
cliente
Servicio
#Producto producto
+Servicio(int precio, int numeroId, String descripcion, String importeIVA, ArrayList<Manufactura> listaDeProductos, ArrayList<Cliente> listaDeClientes, ArrayList<Servicio> listaDeServicios)
+Servicio(int precio, int numeroId, String descripcion)
+Servicio()
+Producto getProducto()
+void vender(Cliente cliente, Servicio servicio)
+int calcularIva(int importeIva)
listaDeProductos
listaDeServicios
<<interface>>
IFacturacion
-void precioFinal(Producto producto, Cliente cliente)
-void aplicarDescuento(Producto producto, Cliente cliente)
-void generarDeuda(Producto producto, Cliente cliente)
<<implementa>>
Empleado
#Empresa empresa
+Empleado(Empresa empresa)
+void vender(Cliente cliente, Manufactura manufactura)
+void venderServicio(Cliente cliente, Servicio servicio)
+void precioFinal(Producto producto, Cliente cliente)
+void aplicarDescuento(Producto producto, Cliente cliente)
+void generarDeuda(Producto producto, Cliente cliente)
empresa
Cliente
#String nombre
#boolean habilitado
#boolean responsableInscripto
#int deuda
#int precioAPagar
+Cliente(String nombre, boolean responsableInscripto, boolean habilitado, boolean efectivo, int deuda)
+Cliente(String nombre, boolean responsableInscripto, boolean efectivo, int deuda)
+boolean clienteHabilitado(Empresa empresa)
+String getNombre()
+void setNombre(String nombre)
+boolean isResponsableInscripto()
+void setResponsableInscripto(boolean responsableInscripto)
+boolean isHabilitado()
+void setHabilitado(boolean estaHabilitado)
+boolean isEfectivo()
+void setEfectivo(boolean efectivo)
+int getDeuda()
+void setDeuda(int deuda)
+int getPrecioAPagar()
+void setPrecioAPagar(int precioAPagar)
listaDeClientes
Empresa
#String nombreDeLaEmpresa
#final int IVA
#static ArrayList<Manufactura> listaDeProductos
#static ArrayList<Cliente> listaDeClientes
#static ArrayList<Servicio> listaDeServicios
+Empresa(String importeIVA, ArrayList<Manufactura> listaDeProductos, ArrayList<Cliente> listaDeClientes, ArrayList<Servicio> listaDeServicios)
+Empresa()
+ArrayList<Manufactura> getListaDeProductos()
+void setListaDeProductos(ArrayList<Manufactura> listaDeProductos)
+ArrayList<Cliente> getListaDeClientes()
+void setListaDeClientes(ArrayList<Cliente> listaDeClientes)
+ArrayList<Servicio> getListaDeServicios()
+void setListaDeServicios(ArrayList<Servicio> listaDeServicios)
Proveedor
-void proveeMateriaPrima(int paquetes)
Program
+static void main(String[] args)


BarajaSpain
~ Naipe[] baraja
~ int cont
+ BarajaSpain()
+ void barajar()
+ String toString()
+ boolean quedanCartas
+ Naipe sacaCarta()
+ Naipe dimeCarta(int i)
1..1
1..1
Brisca
~ BarajaSpain b
~ Naipe[] jugador1
~ Naipe[] jugador2
~ int punJu1
~ int punJu2
~ Naipe muestra
~ int turno
+ Brisca()
+ static void main(String[] args
+ boolean quedaMano()
Naipe
+ final int OROS
+ final int BASTOS
+ final int COPAS
+ final int ESPADAS
+ final int AS
+ final int SOTA
+ final int CABALLO
+ final int REY
- int n
- int p
+ Naipe(int num, int pal)
+ int getCarta()
+ int getPalo()
+ int puntCarta()
+ String toString()
1..1
1..1
muestra


BarajaSpain
~ Naipe[] baraja
~ int cont
+ BarajaSpain()
+ String toString()
Inicio
+ static void main(String[] args
LimiteNaipe
+ void validarCarta(int p, int r
+ int puntCarta()
Dealer
- BarajaSpain bar
+ Dealer(BarajaSpain ba
+ void barajar()
+ boolean quedanCartas
+ Naipe sacaCarta()
+ Naipe dimeCarta(int i)
Brisca
BarajaSpain b
Naipe[] jugador1
Naipe[] jugador2
int punJu1
int punJu2
Naipe getMuestra
int turno
+ Brisca()
+ boolean quedaMano()
+ BarajaSpain getB()
+ Naipe[] getJugador1()
+ Naipe[] getJugador2()
+ int getPunJu1()
+ int getPunJu2()
+ Naipe getMuestra()
+ int getTurno()
Naipe
- int n
- int p
+ Naipe(int n, int p)
+ int getCarta()
+ int getPalo()
+ int puntCarta()
+ String toString()
<<interface>>
INaipe
+ String toString()
+ int puntCarta()
1..1
<<implements>