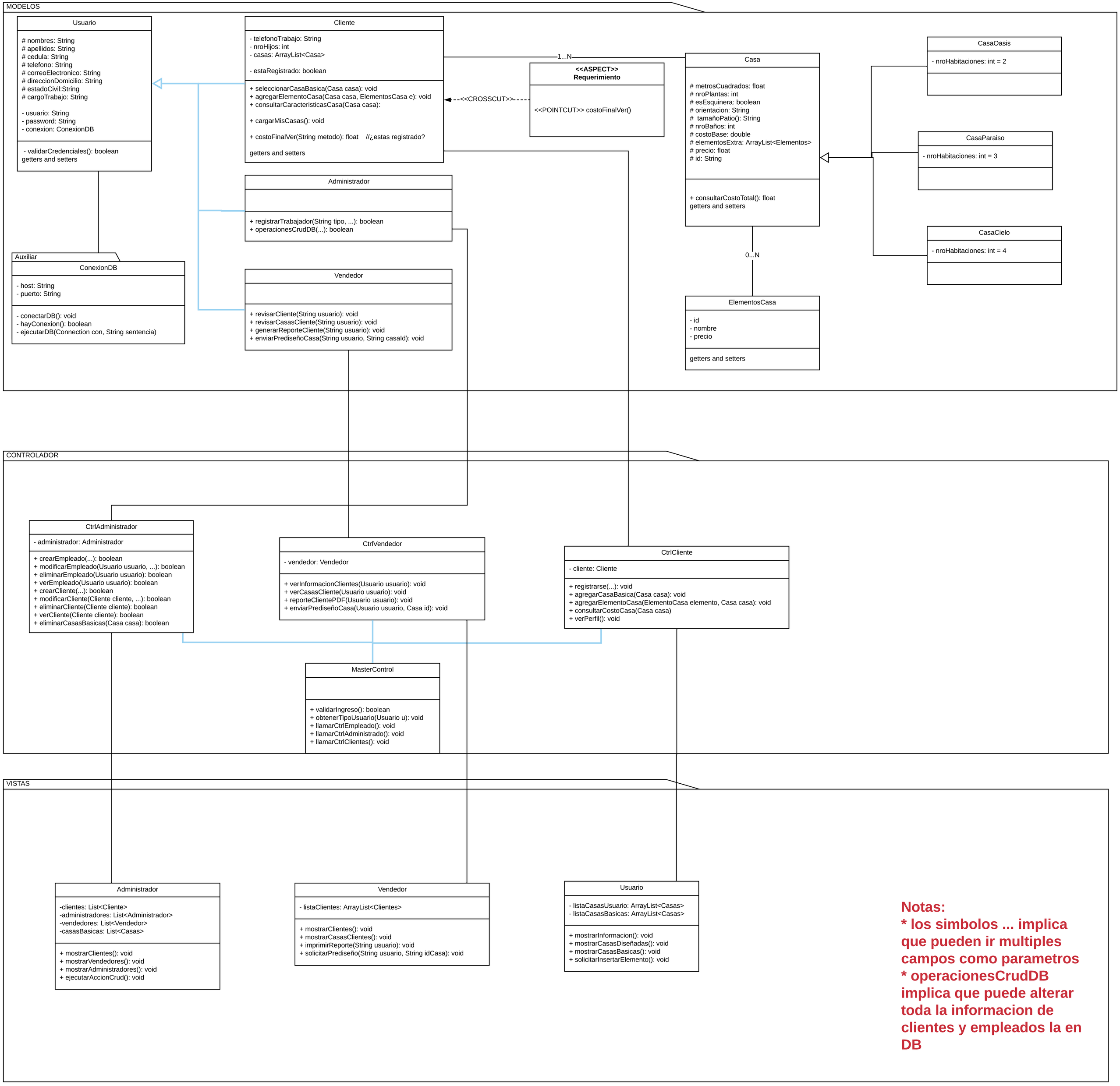
MODELOS
Usuario
nombres: String
apellidos: String
cedula: String
telefono: String
correoElectronico: String
direccionDomicilio: String
estadoCivil:String
cargoTrabajo: String
- usuario: String
- password: String
- conexion: ConexionDB
- validarCredenciales(): boolean
getters and setters
Cliente
- telefonoTrabajo: String
- nroHijos: int
- casas: ArrayList<Casas>
- estaRegistrado: boolean
+ seleccionarCasaBasica(Casa casa): void
+ agregarElementoCasa(Casa casa, ElementosCasa e): void
+ consultarCaracteristicasCasa(Casa casa):
+ cargarMisCasas(): void
+ costoFinalVer(String metodo): float //¿estas registrado?
getters and setters
1...N
<<ASPECT>>
Requerimiento
<<CROSSCUT>>
<<POINTCUT>> costoFinalVer()
Casa
metrosCuadrados: float
nroPlantas: int
esEsquinera: boolean
orientacion: String
tamañoPatio(): String
nroBaños: int
costoBase: double
elementosExtra: ArrayList<Elementos>
precio: float
id: String
+ consultarCostoTotal(): float
getters and setters
CasaOasis
- nroHabitaciones: int = 2
CasaParaiso
- nroHabitaciones: int = 3
CasaCielo
- nroHabitaciones: int = 4
Administrador
+ registrarTrabajador(String tipo, ...): boolean
+ operacionesCrudDB(...): boolean
Auxiliar
ConexionDB
- host: String
- puerto: String
- conectarDB(): void
- hayConexion(): boolean
- ejecutarDB(Connection con, String sentencia)
Vendedor
+ revisarCliente(String usuario): void
+ revisarCasasCliente(String usuario): void
+ generarReporteCliente(String usuario): void
+ enviarPrediseñoCasa(String usuario, String casaId): void
0...N
ElementosCasa
- id
- nombre
- precio
getters and setters
CONTROLADOR
CtrlAdministrador
- administrador: Administrador
+ crearEmpleado(...): boolean
+ modificarEmpleado(Usuario usuario, ...): boolean
+ eliminarEmpleado(Usuario usuario): boolean
+ verEmpleado(Usuario usuario): boolean
+ crearCliente(...): boolean
+ modificarCliente(Cliente cliente, ...): boolean
+ eliminarCliente(Cliente cliente): boolean
+ verCliente(Cliente cliente): boolean
+ eliminarCasasBasicas(Casa casa): boolean
CtrlVendedor
- vendedor: Vendedor
+ verInformacionClientes(Usuario usuario): void
+ verCasasCliente(Usuario usuario): void
+ reporteClientePDF(Usuario usuario): void
+ enviarPrediseñoCasa(Usuario usuario, Casa id): void
CtrlCliente
- cliente: Cliente
+ registrarse(...): void
+ agregarCasaBasica(Casa casa): void
+ agregarElementoCasa(ElementoCasa elemento, Casa casa): void
+ consultarCostoCasa(Casa casa)
+ verPerfil(): void
MasterControl
+ validarIngreso(): boolean
+ obtenerTipoUsuario(Usuario u): void
+ llamarCtrlEmpleado(): void
+ llamarCtrlAdministrado(): void
+ llamarCtrlClientes(): void
VISTAS
Administrador
-clientes: List<Cliente>
-administradores: List<Administrador>
-vendedores: List<Vendedor>
-casasBasicas: List<Casas>
+ mostrarClientes(): void
+ mostrarVendedores(): void
+ mostrarAdministradores(): void
+ ejecutarAccionCrud(): void
Vendedor
- listaClientes: ArrayList<Clientes>
+ mostrarClientes(): void
+ mostrarCasasClientes(): void
+ imprimirReporte(String usuario): void
+ solicitarPrediseño(String usuario, String idCasa): void
Usuario
- listaCasasUsuario: ArrayList<Casas>
- listaCasasBasicas: ArrayList<Casas>
+ mostrarInformacion(): void
+ mostrarCasasDiseñadas(): void
+ mostrarCasasBasicas(): void
+ solicitarInsertarElemento(): void
Notas:
* los simbolos ... implica que pueden ir multiples campos como parametros
* operacionesCrudDB implica que puede alterar toda la informacion de clientes y empleados la en DB