En el diagrama de la interfaz de servicios del sistema, encontramos los siguientes servicios con sus correspondientes operaciones:

* **ServiciosProductos:**
  + ServiciosProductos::ObtenerProductos(t: Técnica, reactivo Boolean) :List(Producto)

Devuelve una lista con los productos reactivos (si Boolean es True) o fungibles (si Boolean es False) registrados en la aplicación utilizados en la técnica t

* + ServiciosProductos::ObtenerInformaciónProducto (id: Integer) : Producto

Dado un producto por su id, devuelve sus características.

* + ServiciosProductos::ObtenerEjemplaresDeProducto(p: Producto) :List(Ejemplar)

Devuelve una lista con los ejemplares que posee actualmente un producto.

* + ServiciosProductos::AñadirProducto(p: Producto) :()

Registra un producto en la aplicación.

* + ServiciosProductos::EditarProducto(p: Producto; np: Producto) :()

Edita un producto existente en la aplicación con las características de np.

* + ServiciosProductos::EliminarProducto(p: Producto) :()

Elimina un producto existente en la aplicación.

* + ServiciosProductos::AñadirEjemplar(e: Ejemplar; p: Producto) :()

Añade el ejemplar e a un producto.

* + ServiciosProductos::EditarEjemplar(e: Ejemplar; ne Ejemplar) :()

Edita el ejemplar e de un producto con las características de ne.

* + ServiciosProductos::RetirarEjemplar(e: Ejemplar) :()

Elimina el ejemplar e de un producto.

* + ServiciosProductos::ConsumirProducto(e: Ejemplar; cantidad: Float) :()

Resta la cantidad consumida del ejemplar del que se ha cogido.

* + ServiciosProductos::AñadirTécnica (t: Técnica) :()

Registra una técnica en la aplicación.

* + ServiciosProductos::EditarTécnica (t: Técnica; nt: Técnica) :()

Edita la técnica t con las características de nt.

* + ServiciosProductos::EliminarTécnica (t: Técnica) :()

Elimina la técnica t de la aplicación.

* **ServiciosProyectos:**
  + ServiciosProyectos::ObtenerAlertas (nivel: Enum(Nivel)) :List(Alerta)

Obtiene todas las alertas que la aplicación ha generado automáticamente. Opcionalmente se podrán filtrar por nivel.

* + ServiciosProyectos::CambiarNivelAlerta (a: Alerta) :()

Cambia el nivel de importancia de una alerta que haya generado el sistema.

* + ServicioProyectos::ObtenerProyectos() :List(Proyecto)

Devuelve una lista con los proyectos registrados en la aplicación

* + ServicioProyectos::ObtenerInformaciónProyecto(id: Integer) : Proyecto

Dada la id de un proyecto, devuelve su información.

* + ServicioProyectos::ObtenerProyectoPorCódigo (código :String) : Proyecto

Dado el código de un proyecto, devuelve su información.

* + ServiciosProductos::AñadirProyecto (p: Proyecto) :()

Registra un producto en la aplicación.

* + ServicioProyectos::EditarProyecto (p : Proyecto) : ()

Edita la información de un proyecto registrado en el sistema.

* + ServicioProyectos::EliminarProyecto (p : Proyecto) : ()

Elimina de un proyecto registrado en el sistema.

* + ServicioProyectos::AñadirPartidaProyecto (pa: Partida, p : Proyecto) : ()

Añade una nueva partida al presupuesto de un proyecto existente.

* + ServicioProyectos::EliminarPartidaProyecto (pa: Partida) : ()

Elimna una partida del presupuesto de un proyecto.

* + ServicioProyectos::EditarPartidaProyecto (pa: Partida) : ()

Edita una partida del presupuesto de un proyecto.

* + ServicioProyectos::DarBajaIntegrante (u :Usuario, p : Proyecto) : ()

Añade a un usuario u a los integrantes del proyecto p.

* + ServicioProyectos::Retirar Integrante (u :Usuario, p : Proyecto) : ()

Elimina a un usuario u de los integrantes del proyecto p.

* **ServiciosPedidos:**
  + ServiciosPedidos::ObtenerProveedores(p: Producto) :List(Proveedor)

Obtiene todas las alertas que la aplicación ha generado automáticamente. Opcionalmente, se podrá filtrar si proveen un producto p en la lista del resultado.

* + ServiciosPedidos::AñadirProveedor(p:Proveedor) : ()

Registra un proveedor en la aplicación

* + ServiciosPedidos::EditarProveedor(p:Proveedor) : ()

Edita un proveedor existente en el sistema.

* + ServiciosPedidos::RetirarProveedor(p:Proveedor) : ()

Marca a un proveedor existente en el sistema como inactivo. De esta manera, no aparecerá disponible para pedidos, pero se conservará en el registro de los pedios.

* + ServiciosPedidos::CrearCreditoConProveedor(p:Proveedor, c: Crédito, pro: Proyecto) : ()

Crea un crédito c para un proyecto pro con el proveedor p.

* + ServiciosPedidos::ObtenerPersonasContactoProveedor(p:Proveedor) : List(PersonaContacto)

Dado un proveedor, devuelve el conjunto de personas de contacto con la empresa de los que se tiene registro.

* + ServiciosPedidos::AñadirPersonaContacto (pc: PersonaContacto, p:Proveedor) :()

Añade una persona de contacto pc a un proveedor p.

* + ServicioProyectos::RetirarPersonaContacto (pc: PersonaContacto) : ()

Eliminar una persona de contacto pc.

* + ServicioProyectos::RealizarPedido(prov: Proveedor, p : List<Producto>, pro: Proyecto, u: Usuario) : ()

Crea un pedido con los productos de la lista p a un proveedor prov a nombre de un proyecto pro o de un usuario u.

* **ServiciosUsuarios:**
  + ServiciosUsuarios:: RegistrarNuevoUsuario(u : Usuario) : ()

Registra a un usuario u en la aplicación.

* + ServiciosUsuarios:: EditarPerfilUsuario(u : Usuario) : ()

Edita la información de un usuario en la aplicación.

* + ServiciosUsuarios:: EliminarUsuario(u : Usuario) : ()

Deshabilita a un usuario en la aplicación.

* + ServiciosUsuarios:: ObtenerUsuarioPorApodo(apodo :String) : Usuario

Dado un apodo, devuelve al usuario cuyo apodo se corresponde con el pasado como parámetro.

* + ServiciosUsuarios:: ObtenerUsuarioPorEmail (email :String) : Usuario

Dado un email, devuelve al usuario cuyo email se corresponde con el pasado como parámetro.

* + ServiciosUsuarios:: ObtenerUsuarios (rol :String) : List(Usuario)

Devuelve una lista con los usuarios registrados en la aplicación. Opcionalmente, se podrán por su rol.