# Vistas de Arquitectura Mi Mutual

* [Mi Mutual Central](#mi-mutual-central)
  + [MiMutual. 1. Contexto](#mimutual.-1.-contexto)
  + [MiMutual. 2. Contenedores](#mimutual.-2.-contenedores)
  + [MiMutual. 3. Dominios](#mimutual.-3.-dominios)
  + [MiMutual. 4. Aplicación](#mimutual.-4.-aplicación)
  + [MiMutual. 4a3. Dependencias](#mimutual.-4a3.-dependencias)
  + [MiMutual. 5. Físico (despliegue)](#mimutual.-5.-físico-despliegue)
  + [MiMutual. 6. Infraestructura](#mimutual.-6.-infraestructura)
  + [MiMutual. 7. Datos. Negocio](#mimutual.-7.-datos.-negocio)
  + [MiMutual. 7a. Datos. Aplicación](#mimutual.-7a.-datos.-aplicación)
  + [MiMutual. 7b. Datos. Relaciones](#mimutual.-7b.-datos.-relaciones)
* [Cotizador Web](#cotizador-web)
  + [Cotizador. 1. Contexto](#cotizador.-1.-contexto)
  + [Cotizador. 2. Contenedores](#cotizador.-2.-contenedores)
  + [Cotizador. 4. Aplicación](#cotizador.-4.-aplicación)
  + [Cotizador. 4a. Aplicación. Servicios](#cotizador.-4a.-aplicación.-servicios)
  + [Cotizador. 4a. Dependencias](#cotizador.-4a.-dependencias)
  + [upd](#upd)
  + [Cotizador. 7. Datos. Negocio](#cotizador.-7.-datos.-negocio)

# Mi Mutual Central

## MiMutual. 1. Contexto



Figure 1: Diagram: MiMutual. 1. Contexto

## Contexto Mi Mutual Central

El sistema principal de fondo Mi Mutual Central es la composición de las funciones de negocio de la Unidad de Solidaridad de Coomeva. Las funciones de negocio referidas, como Gestión Beneficiarios, Certificados, Gestión Beneficiarios, aparecen dentro del componente principal en la imagen.

Este entregable documenta los diferentes módulos y componentes que hacen parte de la estructura de una aplicación en Angular 12 y como es su interacción para conformar una arquitectura robusta y escalable para aplicaciones de gran tamaño.

Las librerías Spring Boot Security y Spring Boot Oauth2 proveen características de seguridad entre Vista (Angular 2) y Controlador. Estas son responsables de que únicamente permita el acceso si se está autenticado. Además, para realizar el proceso de autenticación se delega a la aplicación SISPRO (Coomeva) que funciona como un servidor de autenticación.

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **app: Mi Mutual Central** | application-component | Antes SIPAS, Mi Mutual es una aplicación web compuesta por distintos módulos de software con arreglo a todas las actividades necesarias que soportan la operación de los productos y servicios que ofrece la Unidad de Solidaridad y Seguros de la Cooperativa. | *modulo:* mimutual |
| **Administración facturación y recaudo** | application-function | Administración de la facturación y recaudo diario de los productos |  |
| **Autorizaciones** | application-function | Autorizaciones: Administración de peticiones de autorización y sus correspondientes aprobaciones mediante el servicio del flujo de procesos |  |
| **Certificados** | application-function | Certificados: Permite la generación de los certificados de valores de protección y contribuciones pagadas, de retención en la fuente, de pagos de perseverancia y de cobertura de auxilio funerario. |  |
| **Configuración factores cálculos, contribuciones** | application-function | 1. Configuración o parametrización de factores para realizar los cálculos de las contribuciones de los asociados a la Cooperativa para cada uno de los productos adquiridos. |  |
| **Gestión Beneficiarios** | application-function | Gestión de Beneficiarios: Permite administrar la información relacionada con los beneficiarios del Asociado, permitiendo ejecutar operaciones de consulta, inserción y modificación. |  |
| **Gestión Reclamaciones** | application-function | Gestión de Reclamaciones (Indemnización): Permite realizar la gestión, seguimiento y pago o negación de las diferentes reclamaciones de acuerdo a las coberturas y los productos que se encuentren dentro del portafolio del Asociado. |  |
| **Gestión Usuarios** | application-function | Gestión de Usuarios: Administración de la información relacionada con los usuarios del sistema. Este componente se comunica con el servicio unificado de autenticación y autorización que devuelve los permisos que un usuario posee sobre las opciones que proporciona el sistema. |  |
| **Gestión fondo mutual y auxilio funerario** | application-function | Gestión de productos del fondo mutual y auxilio funerario que involucran a sus coberturas |  |
| **Interoperabilidad entre sistemas Coomeva** | application-function | Integración con otros sistemas para facilitar los procesos de vinculación, retiro, reactivación o fallecimiento de asociados. |  |
| **Seguridad** | application-function |  |  |
| **Simuladores** | application-function | Simuladores: Funcionalidades que permiten generar las simulaciones de los diferentes planes o modificaciones (incrementos y disminuciones) a los productos del Asociado. |  |
| **Autorizaciones** | application-service | Autorizaciones: Administración de peticiones de autorización y sus correspondientes aprobaciones usando el servicio del flujo de procesos. |  |
| **Certificados** | application-service | Certificados: Permite la generación de los certificados de valores de protección y contribuciones pagadas, de retención en la fuente, de pagos de perseverancia y de cobertura de auxilio funerario. |  |
| **Configuración** | application-service | Configuración o parametrización de factores para realizar los cálculos de las contribuciones de los asociados a la Cooperativa para cada uno de los productos adquiridos. |  |
| **Facturación y Recaudo** | application-service | Administración de la facturación y recaudo diario de los productos. |  |
| **Gestión de Beneficiarios** | application-service | Gestión de Beneficiarios: Permite administrar la información relacionada con los beneficiarios del Asociado, permitiendo ejecutar operaciones de consulta, inserción y modificación. |  |
| **Gestión de Productos** | application-service | Gestión de productos del fondo mutual y auxilio funerario que involucran lo relacionado a las siguientes coberturas: \* Fondo de Solidaridad: Incapacidades temporales, Incapacidades Permanentes (total, parcial), Perseverancia 60, 62, 65, 70 años, Perseverancias Anticipadas, Fallecimiento Asociado (Auxilio por muerte), Desempleo, Disminución de ingresos y enfermedades graves; Rentas por hospitalización, Enfermedades de Alto Costo, Pólizas de seguros personales y patrimoniales, Planes educativos, Segunda opinión médica, Asistencias. \* Auxilio Funerario: Fallecimiento de familiares directos (inscritos) del Asociado. |  |
| **Gestión de Reclamos** | application-service | Gestión de Reclamaciones (Indemnización): Permite realizar la gestión, seguimiento y pago o negación de las diferentes reclamaciones de acuerdo a las coberturas y los productos que se encuentren dentro del portafolio del Asociado |  |
| **Gestión de Usuarios** | application-service | Gestión de Usuarios: Administración de la información relacionada con los usuarios del sistema. Este componente se comunica con el servicio unificado de autenticación y autorización que devuelve los permisos que un usuario posee sobre las opciones que proporciona el sistema. |  |
| **Simuladores** | application-service | Simuladores: Funcionalidades que permiten generar las simulaciones de los diferentes planes o modificaciones (incrementos y disminuciones) a los productos del Asociado. |  |
| **Unidad de Solidaridad y Seguros** | business-function | Unidad de Solidaridad y Seguros de la Cooperativa |  |
| **Restricciones de Arquitectura** | constraint | ## Restricciones de Arquitectura (Atributos)1. Disponibilidad. Se requiere que el sistema esté disponible 7 X 24, el servicio prestado al cliente no se limita a horarios de oficina pues las compras pueden darse en cualquier momento1. Escalabilidad. Se requiere que el sistema pueda llegar a atender hasta 1.000 clientes, para esto se requiere que el sistema se pueda extender horizontalmente de tal manera que pueda tener instalado en varios servidores para atender esta cantidad de usuarios. Todas las aplicaciones desarrolladas podrán ser escaladas horizontalmente para atender la demanda relacionada con el crecimiento de la empresa.1. Reutilización. Se requiere que el sistema permita reutilizar sus componentes para prestar el mismo servicio a otras aplicaciones de la compañía. Para esto se va a desarrollar la aplicación utilizando servicios, separados y con asignación de responsabilidades, propias, de tal manera de que, si se requiere exponer servicios web sobre estas funcionalidades, no requiere cambios en la aplicación.1. Autenticación. Autenticación es el proceso para determinar que alguien o un sistema es quien dice ser. Uso de estándar Oauth2 y JSON Web Token – JWT, para gestión de autenticación de servicios de la aplicación.1. Autorización. Autorización se refiere a la validación que realiza un sistema para determinar si un usuario puede usar cierta funcionalidad. Uso de API de seguridad de Spring (spring-security) + Oauth21. Interoperabilidad – Movilidad. Interoperabilidad se refiere a la habilidad de un sistema de interactuar y comunicarse con sistemas heterogéneos a través de interfaces completamente definidas. Uso de estándar de web services REST + JSON.1. Facilidad de Uso. Se refiere a la facilidad con que las personas pueden utilizar el sistema porque facilitan la lectura de los textos, descargan rápidamente la información y presentan funciones y menús sencillos, por lo que el usuario encuentra satisfechas sus consultas y cómodo su uso.1. Verificación (QA). Es la capacidad del producto software que hace posible que el software modificado sea probado. |  |
| **Características Funcionales** | requirement | ## Características Funcionales Mi Mutual1. Gestión de productos del fondo mutual y auxilio funerario que involucran a sus coberturas1. Administración de la facturación y recaudo diario de los productos1. Gestión de Reclamaciones (Indemnización): Permite realizar la gestión, seguimiento y pago o negación de las diferentes reclamaciones de acuerdo a las coberturas y los productos que se encuentren dentro del portafolio del Asociado.1. Gestión de Beneficiarios: Permite administrar la información relacionada con los beneficiarios del Asociado, permitiendo ejecutar operaciones de consulta, inserción y modificación.1. Gestión de Usuarios: Administración de la información relacionada con los usuarios del sistema. Este componente se comunica con el servicio unificado de autenticación y autorización que devuelve los permisos que un usuario posee sobre las opciones que proporciona el sistema.1. Integración con otros sistemas para facilitar los procesos de vinculación, retiro, reactivación o fallecimiento de asociados.1. Configuración o parametrización de factores para realizar los cálculos de las contribuciones de los asociados a la Cooperativa para cada uno de los productos adquiridos. |  |

## MiMutual. 2. Contenedores



Figure 2: Diagram: MiMutual. 2. Contenedores

La organización de componentes utilizada Mi Mutual, impulsada por Spring Web, antepone como interfaz de uso un API REST. La interfaz se articula con tres componentes utilitarios: Controller, Service y Repository, los cuales están mediados por el componente misional, Mi Mutual, en la imagen.

Esta decisión de organización de los componentes de Mi Mutual, incluyendo al misional del mismo nomnbre, permite estructurar la aplicación de una manera ordenada y, en línea con las restricciones de arquitectura exigidas al sistema, facilita la efectividad de las extensiones y el mantenimiento.

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **Controlador Mi Mutual** | application-component | Los componentes de este tipo se encargan de controlar los servicios rest de la aplicación, además en estos componentes se define la forma como se reciben y envían los datos de los servicios rest y la seguridad de cada uno de los métodos. | *modulo:* mimutual |
| **Registro Servicios: eureka** | application-component | Eureka: Contiene todas las funcionalidades relacionadas con registrar y localizar microservicios existentes, informar de su localización, su estado y datos relevantes de cada uno de ellos. | *modulo:* mimutual |
| **Repositorio Mi Mutual** | application-component | Antes SIPAS, Mi Mutual es una aplicación web compuesta por distintos módulos de software con arreglo a todas las actividades necesarias que soportan la operación de los productos y servicios que ofrece la Unidad de Solidaridad y Seguros de la Cooperativa.Para el manejo de la persistencia de datos se utilizará Spring Data el cual se apoya en la especificación de JPA y en la implementación de HIBERNATE además de complementar esta capa de persistencia con nuevas funcionalidades que facilitan el acceso a datos. | *modulo:* mimutual |
| **Ruteador (proxy): gateway** | application-component | Gateway: Contiene todas las funcionalidades relacionadas con un proxy inverso que reenvía las llamadas relevantes a otros servicios. | *modulo:* mimutual |
| **app: Cotizador Web** | application-component | pkg: MiMutualWeb | *modulo:* cotizador |
| **app: Implementación de Servicios** | application-component | Los componentes de este tipo se encargan de controlar y almacenar toda la lógica del negocio, validaciones y todo lo referente a procesamiento de datos. | *modulo:* mimutual |
| **app: Mi Mutual Central** | application-component | Antes SIPAS, Mi Mutual es una aplicación web compuesta por distintos módulos de software con arreglo a todas las actividades necesarias que soportan la operación de los productos y servicios que ofrece la Unidad de Solidaridad y Seguros de la Cooperativa. | *modulo:* mimutual |
| **API Mi Mutual** | application-interface |  | *modulo:* mimutual |

## MiMutual. 3. Dominios



Figure 3: Diagram: MiMutual. 3. Dominios

La división por dominios busca facilitar la administración los servicios de la plataforma Mi Mutual que son comunes entre aplicaciones de Mi Mutual, tales como Asociados, Reclamaciones, Protecciones y otros servicios trasversales como Utilidades, Reglas de negocio, Procesos de negocio (BPM), Auditoria, Flujos de trabajo:flowable.

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **Flujo Trabajo: flowable** | application-component | Contiene todas las funcionalidades relacionadas con el motor de BPM Flowable, como gestión de tareas, instancias de nuevas procesos y asignación de tareas. | *modulo:* mimutual |
| **Integración** | application-component | Contiene todas las funcionalidades relacionadas con integraciones a otros servicios y otras bases de datos. | *modulo:* mimutual |
| **Procesos negocio** | application-component | Contiene todas las funcionalidades relacionadas con los flujos de JBPM, actualmente se hace solo para acceso a datos de la base de datos de JBPM. | *modulo:* mimutual |
| **Reglas negocio** | application-component | Contiene todas las funcionalidades relacionadas con la validación de reglas usadas en otros microservicios. | *modulo:* mimutual |
| **SISPRO** | application-component | Contiene todas las funcionalidades relacionadas con la autenticación y autorización al sistema Mi mutual (Este componente se adopta a la arquitectura de microservicios de MiMutual) | *modulo:* mimutual |
| **Utilidades** | application-component | Contiene todas las funcionalidades útiles y trasversales a los microservicios, como envió de correos, generación de archivos Excel, PDF desde Jasper y consulta de parámetros. | *modulo:* mimutual |
| **app: Asociados** | application-component | Contiene todas las funcionalidades relacionadas con consulta y creación de asociados y beneficiarios. | *modulo:* mimutual |
| **app: Auditoria** | application-component | Contiene todas las funcionalidades relacionadas con el almacenamiento de la auditoria de las peticiones de la aplicación. | *modulo:* mimutual |
| **app: Eureka admin** | application-component | Contiene todas las funcionalidades relacionadas con registrar y localizar microservicios existentes, informar de su localización, su estado y datos relevantes de cada uno de ellos. | *modulo:* mimutual |
| **app: Gateway** | application-component | Contiene todas las funcionalidades relacionadas con un proxy inverso que reenvía las llamadas relevantes a otros servicios. | *modulo:* mimutual |
| **app: Identidades** | application-component | Contiene todas las funcionalidades relacionadas con la gestión de los archivos de propiedades de los microservicios (Esta en construcción y no se ha integrado). | *modulo:* mimutual |
| **app: Mi Mutual Central** | application-component | Antes SIPAS, Mi Mutual es una aplicación web compuesta por distintos módulos de software con arreglo a todas las actividades necesarias que soportan la operación de los productos y servicios que ofrece la Unidad de Solidaridad y Seguros de la Cooperativa. | *modulo:* mimutual |
| **app: Protecciones** | application-component | Contiene todas las funcionalidades relacionadas con la gestión y configuración de productos y protecciones. | *modulo:* mimutual |
| **app: Reclamaciones** | application-component | Contiene todas las funcionalidades relacionadas con la gestión de reclamaciones, liquidaciones y pagos. | *modulo:* mimutual |
| **app: Secuencias: zipkin** | application-component | Contiene todas las funcionalidades relacionadas con la generación de IDs para la trazabilidad de los logs. | *modulo:* mimutual |

## MiMutual. 4. Aplicación



Figure 4: Diagram: MiMutual. 4. Aplicación

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **Controlador Mi Mutual** | application-component | Los componentes de este tipo se encargan de controlar los servicios rest de la aplicación, además en estos componentes se define la forma como se reciben y envían los datos de los servicios rest y la seguridad de cada uno de los métodos. | *modulo:* mimutual |
| **Mi Mutual Presentación** | application-component |  |  |
| **Repositorio Mi Mutual** | application-component | Antes SIPAS, Mi Mutual es una aplicación web compuesta por distintos módulos de software con arreglo a todas las actividades necesarias que soportan la operación de los productos y servicios que ofrece la Unidad de Solidaridad y Seguros de la Cooperativa.Para el manejo de la persistencia de datos se utilizará Spring Data el cual se apoya en la especificación de JPA y en la implementación de HIBERNATE además de complementar esta capa de persistencia con nuevas funcionalidades que facilitan el acceso a datos. | *modulo:* mimutual |
| **Simulador** | application-component |  |  |
| **Vinculación** | application-component |  |  |
| **app: Cotización y Ventas** | application-component |  |  |
| **app: Cotizador Web** | application-component | pkg: MiMutualWeb | *modulo:* cotizador |
| **app: Implementación de Servicios** | application-component | Los componentes de este tipo se encargan de controlar y almacenar toda la lógica del negocio, validaciones y todo lo referente a procesamiento de datos. | *modulo:* mimutual |
| **app: Mi Mutual Central** | application-component | Antes SIPAS, Mi Mutual es una aplicación web compuesta por distintos módulos de software con arreglo a todas las actividades necesarias que soportan la operación de los productos y servicios que ofrece la Unidad de Solidaridad y Seguros de la Cooperativa. | *modulo:* mimutual |
| **Interfaz transporte** | application-interface | Feign Client.Integración con otros sistemas para facilitar los procesos de vinculación, retiro, reactivación o fallecimiento de asociados. |  |
| **Autorizaciones** | application-service | Autorizaciones: Administración de peticiones de autorización y sus correspondientes aprobaciones usando el servicio del flujo de procesos. |  |
| **Certificados** | application-service | Certificados: Permite la generación de los certificados de valores de protección y contribuciones pagadas, de retención en la fuente, de pagos de perseverancia y de cobertura de auxilio funerario. |  |
| **Configuración** | application-service | Configuración o parametrización de factores para realizar los cálculos de las contribuciones de los asociados a la Cooperativa para cada uno de los productos adquiridos. |  |
| **Facturación y Recaudo** | application-service | Administración de la facturación y recaudo diario de los productos. |  |
| **Gestión de Beneficiarios** | application-service | Gestión de Beneficiarios: Permite administrar la información relacionada con los beneficiarios del Asociado, permitiendo ejecutar operaciones de consulta, inserción y modificación. |  |
| **Gestión de Productos** | application-service | Gestión de productos del fondo mutual y auxilio funerario que involucran lo relacionado a las siguientes coberturas: \* Fondo de Solidaridad: Incapacidades temporales, Incapacidades Permanentes (total, parcial), Perseverancia 60, 62, 65, 70 años, Perseverancias Anticipadas, Fallecimiento Asociado (Auxilio por muerte), Desempleo, Disminución de ingresos y enfermedades graves; Rentas por hospitalización, Enfermedades de Alto Costo, Pólizas de seguros personales y patrimoniales, Planes educativos, Segunda opinión médica, Asistencias. \* Auxilio Funerario: Fallecimiento de familiares directos (inscritos) del Asociado. |  |
| **Gestión de Reclamos** | application-service | Gestión de Reclamaciones (Indemnización): Permite realizar la gestión, seguimiento y pago o negación de las diferentes reclamaciones de acuerdo a las coberturas y los productos que se encuentren dentro del portafolio del Asociado |  |
| **Gestión de Usuarios** | application-service | Gestión de Usuarios: Administración de la información relacionada con los usuarios del sistema. Este componente se comunica con el servicio unificado de autenticación y autorización que devuelve los permisos que un usuario posee sobre las opciones que proporciona el sistema. |  |
| **SS02.acumulados - portafolio** | application-service |  |  |
| **SS02.aportes - estatutarios** | application-service |  |  |
| **SS02.distribuciones** | application-service |  |  |
| **SS02.notas - aclaratorias** | application-service |  |  |
| **SS02.porcentaje - cuota** | application-service |  |  |
| **SS02.protecciones - mim - actuaria** | application-service | Servicios de aplicación central que el Cotizador Web usa. Protección. |  |
| **SS02.protecciones- desmemebracion - accidente** | application-service | Servicios de aplicación central que el Cotizador Web usa. Protección. |  |
| **SS02.proveedor** | application-service |  |  |
| **SS02.proyeccion** | application-service |  |  |
| **SS02.proyeccion** | application-service |  |  |
| **SS02.proyecto - vida** | application-service |  |  |
| **SS02.proyecto - vida** | application-service |  |  |
| **SS02.relacion - planes** | application-service |  |  |
| **SS02.relacion - planes** | application-service |  |  |
| **SS02.reporte - cotizacion** | application-service | Servicios de aplicación central que el Cotizador Web usa. Reportes. |  |
| **SS02.reporte - estado - cotizacion** | application-service | Servicios de aplicación central que el Cotizador Web usa. Cotización. |  |
| **SS02.tipo - plan** | application-service |  |  |
| **SS02.util - archivos** | application-service |  |  |
| **SS02.util - municipios** | application-service |  |  |
| **Simuladores** | application-service | Simuladores: Funcionalidades que permiten generar las simulaciones de los diferentes planes o modificaciones (incrementos y disminuciones) a los productos del Asociado. |  |
| **Analistas** | business-role | Analistas y auxiliares de servicio regional y nacional, agentes del centro de contacto, auditores médicos, analistas de operaciones (aseguramiento y facturación) y jefes. |  |
| **Asesores** | business-role | Asesores integrales |  |
| **Auxiliares servicio** | business-role | Analistas y auxiliares de servicio regional y nacional, agentes del centro de contacto, auditores médicos, analistas de operaciones (aseguramiento y facturación) y jefes. |  |
| **Fidelización** | business-role | Ejecutivos de Fidelización |  |
| **Profundización** | business-role | Ejecutivos de Profundización |  |
| **Unidad de Solidaridad y Seguros** | grouping | La Unidad de Solidaridad y Seguros cuenta con un software integrado para su core de negocio denominado SIPAS (Sistema de Previsión, Asistencia y Solidaridad) |  |
| **Servicio de Almacenamiento de Datos** | technology-service |  |  |
| **Servicio de Red** | technology-service |  |  |
| **Servicio de archivos** | technology-service |  |  |

## MiMutual. 4a3. Dependencias



Figure 5: Diagram: MiMutual. 4a3. Dependencias

### Paquetes y Dependencias Mi Mutual

La estructura está basada en spring boot y lenguaje de programación JAVA 8, conformada por componentes de aplicación y administración del ciclo de vida de los objetos.

Nombrado de paquetes para los servicios

* MiMutualWeb
* MiMutualProtecciones
* MiMutualReclamaciones
* MiMutualAsociados
* MiMutualUtilidades
* MiMutualBPM
* MiMutualReglas
* MiMutualIntegraciones
* MiMutualAuditoria
* Mi<utualFLowable
* MiMutualSpringCloud

### Entorno de Desarrollo

Para la etapa de desarrollo la aplicación estará configurada para levantar un servidor Tomcat embebido el cual se encuentra configurado el pom.xml y el cual permite trabajar de forma mucho más ágil.

Para el despliegue entre ambientes se manejará maven profiles con el fin de agregar las configuraciones de cada uno de estos.

El código fuente está alojado en un repositorio de Coomeva.

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **app: Mi Mutual Central** | application-component | Antes SIPAS, Mi Mutual es una aplicación web compuesta por distintos módulos de software con arreglo a todas las actividades necesarias que soportan la operación de los productos y servicios que ofrece la Unidad de Solidaridad y Seguros de la Cooperativa. | *modulo:* mimutual |
| **pkg: Config** | application-component | config: Almacenan todas las clases para la configuración del proyecto Spring. |  |
| **pkg: Controller** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. |  |
| **pkg: DTO** | application-component | dto: Almacenan todas las clases (pojos) para manejo de datos. |  |
| **pkg: Model** | application-component | model: Almacenan todas las clases (entities) que constituyen el modelo de datos. |  |
| **pkg: Reglas** | application-component | dto: Almacenan todas las clases (pojos) para manejo de datos. |  |
| **pkg: Repository** | application-component | repository: Almacenan todas las interfaces y clases que constituyen el acceso adatos. |  |
| **pkg: Security** | application-component | security: Almacenan todas las clases que permiten la configuración de seguridad de la aplicación. |  |
| **pkg: Service** | application-component | service: Almacenan todas las interfaces que constituyen la lógica de negocio. |  |
| **pkg: Service Impl** | application-component | service.impl: Almacenan todas las clases que implementan la funcionalidad de las interfaces de service que constituyen la lógica de negocio. |  |
| **pkg: Spring Cloud** | application-component | dto: Almacenan todas las clases (pojos) para manejo de datos. |  |
| **pkg: Util** | application-component | util: Almacenan todas las clases de utilería para la aplicación. |  |
| **pkg: Web** | application-component | dto: Almacenan todas las clases (pojos) para manejo de datos. |  |

## MiMutual. 5. Físico (despliegue)



Figure 6: Diagram: MiMutual. 5. Físico (despliegue)

Especificaciones de despliegue Mi Mutual, 2023, componente central.

* Estándares para el manejo de servicios REST sobre HTTP 1.1
* Tecnologías para el backend: Java 8 con Spring Boot2.1.4
* Acceso a Datos: Spring Data 2.1.4
* Seguridad de las API: Spring Security + Oauth2.0
* Plataforma de despliegue Backend: Tomcat Spring boot
* Tecnologías para el frontend: Angular 12
* Librería de Estilos: Bootstrap 4
* Servidor web (HTTP 1.1): Apache 2.X

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **app: Asociados** | application-component | Contiene todas las funcionalidades relacionadas con consulta y creación de asociados y beneficiarios. | *modulo:* mimutual |
| **app: Eureka admin** | application-component | Contiene todas las funcionalidades relacionadas con registrar y localizar microservicios existentes, informar de su localización, su estado y datos relevantes de cada uno de ellos. | *modulo:* mimutual |
| **app: Gateway** | application-component | Contiene todas las funcionalidades relacionadas con un proxy inverso que reenvía las llamadas relevantes a otros servicios. | *modulo:* mimutual |
| **app: Identidades** | application-component | Contiene todas las funcionalidades relacionadas con la gestión de los archivos de propiedades de los microservicios (Esta en construcción y no se ha integrado). | *modulo:* mimutual |
| **app: Implementación de Servicios** | application-component | Los componentes de este tipo se encargan de controlar y almacenar toda la lógica del negocio, validaciones y todo lo referente a procesamiento de datos. | *modulo:* mimutual |
| **app: Mi Mutual Central** | application-component | Antes SIPAS, Mi Mutual es una aplicación web compuesta por distintos módulos de software con arreglo a todas las actividades necesarias que soportan la operación de los productos y servicios que ofrece la Unidad de Solidaridad y Seguros de la Cooperativa. | *modulo:* mimutual |
| **app: Protecciones** | application-component | Contiene todas las funcionalidades relacionadas con la gestión y configuración de productos y protecciones. | *modulo:* mimutual |
| **app: Reclamaciones** | application-component | Contiene todas las funcionalidades relacionadas con la gestión de reclamaciones, liquidaciones y pagos. | *modulo:* mimutual |
| **app: Secuencias: zipkin** | application-component | Contiene todas las funcionalidades relacionadas con la generación de IDs para la trazabilidad de los logs. | *modulo:* mimutual |
| **Directorio: eureka: tomcat** | system-software |  |  |
| **Entorno Java: JRE 1.8** | system-software |  |  |
| **Gateway: tomcat** | system-software |  |  |
| **Identidad: spring cloud security** | system-software |  |  |
| **Secuencias: zipking: tomcat** | system-software |  |  |
| **Servicios: tomcat** | system-software |  | *modulo:* mimutual |

## MiMutual. 6. Infraestructura



Figure 7: Diagram: MiMutual. 6. Infraestructura

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **app: Mi Mutual Central** | application-component | Antes SIPAS, Mi Mutual es una aplicación web compuesta por distintos módulos de software con arreglo a todas las actividades necesarias que soportan la operación de los productos y servicios que ofrece la Unidad de Solidaridad y Seguros de la Cooperativa. | *modulo:* mimutual |
| **Spring cloud security** | node | Se implementará Spring Boot Security y Spring Boot Oauth2 las cuales proveen una capa básica de seguridad entre Vista (Angular 2) y Controlador, obligando a que únicamente permita el acceso si se está autenticado si lo requiere, además para realizar el proceso de autenticación se utilizara la aplicación SISPRO (Coomeva) la cual funciona como un servidor de autenticación.La validación de roles se realizará a nivel de peticiones en el api rest según corresponda, siempre y cuando sea necesario, también se tendrán en cuenta otras validaciones como:• Las credenciales que proporcionó no son válidas.• El usuario esta deshabilitado.• Las credenciales de usuario han caducado.• La cuenta de usuario ha caducado.• La cuenta de usuario está bloqueada. |  |
| **DB2 AS400: SIPASDB** | system-software | Las bases de datos a utilizar son: \* DB2 AS400: SIPASDB, \* Sql Server (…), \* Oracle (…) |  |
| **Oracle DB: (…)** | system-software | Las bases de datos a utilizar son: \* DB2 AS400: SIPASDB, \* SQL Server (…), \* Oracle (…) |  |
| **SQL Server: (…)** | system-software | Las bases de datos a utilizar son: \* DB2 AS400: SIPASDB, \* SQL Server (…), \* Oracle (…) |  |
| **Servicio de Almacenamiento de Datos** | technology-service |  |  |
| **Servicio de Red** | technology-service |  |  |
| **Servicio de archivos** | technology-service |  |  |

## MiMutual. 7. Datos. Negocio

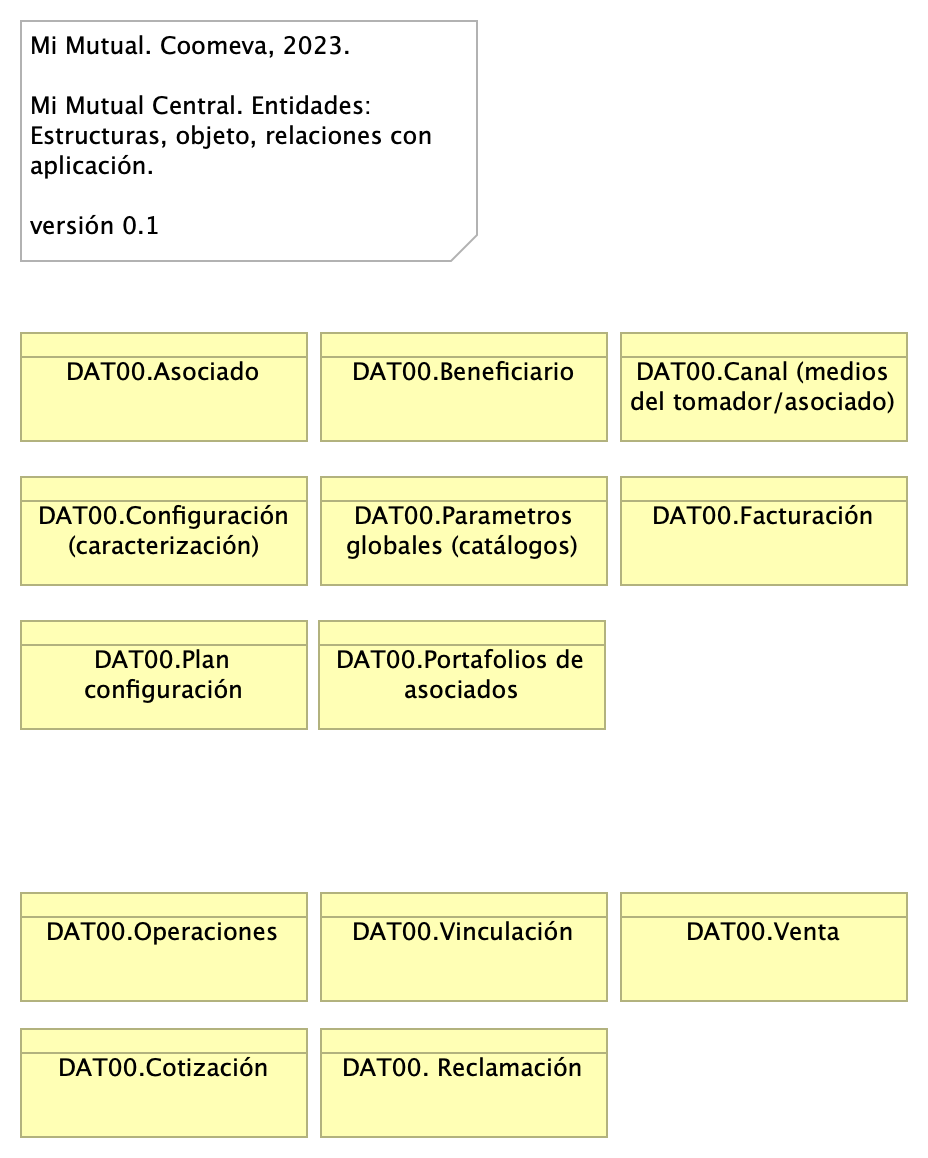


Figure 8: Diagram: MiMutual. 7. Datos. Negocio

### Entidades de Negocio Mi Mutual

Dominios de datos de negocio. Entidades independiente de la plataforma y de la tecnología.

* Configuración (caracterización de productos, plan)
* Plan (producto pólizas seguros)
* Canal (medios del tomador/asociado)
* Parametros globales (catálogos)
* Portafolio de asociado
* Asociado
* Facturación
* Beneficiario

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **DAT00. Reclamación** | business-object | Reclamacion es cuando el asociado desea que le paguen una cobertura, Ejemplo cumplio perseverancia ya tiene 60 años y cobra. |  |
| **DAT00.Asociado** | business-object |  |  |
| **DAT00.Beneficiario** | business-object |  |  |
| **DAT00.Canal (medios del tomador/asociado)** | business-object |  |  |
| **DAT00.Configuración (caracterización)** | business-object | Caracterización de productos, planes, parámetros |  |
| **DAT00.Cotización** | business-object |  |  |
| **DAT00.Facturación** | business-object | Factura la genera COOMEVA. |  |
| **DAT00.Operaciones** | business-object |  |  |
| **DAT00.Parametros globales (catálogos)** | business-object |  |  |
| **DAT00.Plan configuración** | business-object | Plan de configuración: producto pólizas seguros. |  |
| **DAT00.Portafolios de asociados** | business-object |  |  |
| **DAT00.Venta** | business-object |  |  |
| **DAT00.Vinculación** | business-object |  |  |

## MiMutual. 7a. Datos. Aplicación



Figure 9: Diagram: MiMutual. 7a. Datos. Aplicación

Estructuras de datos específicas a la plataforma. Modelo de negocio para las aplicaciones.

Modelo físico facilitadas por Coomeva, corte del 2 de mayo de 2022. Contiene las estructuras de configuración de fondo, cliente, planes, cobertura y planes de coberturas.

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **DAT01.ADICIONAL PLAN COBERTURA** | data-object |  |  |
| **DAT01.ASISTENCIA PLAN COBERTURA** | data-object |  |  |
| **DAT01.ASISTENCIA PLAN COBERTURA DETALLE** | data-object |  |  |
| **DAT01.BENEFICIARIO ASISTENCIA** | data-object |  |  |
| **DAT01.BENEFICIO PREEXISTENCIA** | data-object |  |  |
| **DAT01.CAMPANA RELACION PLAN COBERTURA** | data-object |  |  |
| **DAT01.CANAL** | data-object |  |  |
| **DAT01.CANAL EVENTO** | data-object |  |  |
| **DAT01.CANAL VENTA EXCLUSION** | data-object |  |  |
| **DAT01.CANAL VENTA EXCLUSION COBERTURA** | data-object |  |  |
| **DAT01.CICLO FACTURACION CONFIG MOV** | data-object |  |  |
| **DAT01.CLIENTE** | data-object |  |  |
| **DAT01.COBERTURA** | data-object |  |  |
| **DAT01.COBERTURA BENEFICIARIO COBERTURA** | data-object |  |  |
| **DAT01.COBERTURA BENEFICIARIO PAGO** | data-object |  |  |
| **DAT01.CONDICION PAGO ANTIGUEDAD** | data-object |  |  |
| **DAT01.CONDICION PAGO EVENTO** | data-object |  |  |
| **DAT01.CONDICION PLAN** | data-object |  |  |
| **DAT01.CONDICION VENTA** | data-object |  |  |
| **DAT01.CONDICIONES** | data-object |  |  |
| **DAT01.CONFIGURACION DIAGNOSTICOS** | data-object |  |  |
| **DAT01.CONFIGURACION MOV DETALLE** | data-object |  |  |
| **DAT01.CONFIGURACION MOVIMIENTO** | data-object |  |  |
| **DAT01.CONTROL ARE TECNICA** | data-object |  |  |
| **DAT01.CONTROL CUMULO** | data-object |  |  |
| **DAT01.CUMULO** | data-object |  |  |
| **DAT01.CUMULOS COBERTURA** | data-object |  |  |
| **DAT01.CUMULOS PLAN COBERTURA** | data-object |  |  |
| **DAT01.DEDUCIBLE** | data-object |  |  |
| **DAT01.DESMEMBRACION ACCIDENTE** | data-object |  |  |
| **DAT01.DESMEMBRACION ACCIDENTE PLAN COBERTURA** | data-object |  |  |
| **DAT01.DOCUMENTO REQUISITO** | data-object |  |  |
| **DAT01.DOCUMENTOS SOLICITUD EVENTO** | data-object |  |  |
| **DAT01.ENFERMEDAD GRAVE** | data-object |  |  |
| **DAT01.ENFERMEDAD GRAVE PLAN COBERTURA** | data-object |  |  |
| **DAT01.ESTADO ASOCIADO** | data-object |  |  |
| **DAT01.EVENTO COBERTURA** | data-object |  |  |
| **DAT01.EXCLUSION** | data-object |  |  |
| **DAT01.EXCLUSION COBERTURA** | data-object |  |  |
| **DAT01.EXCLUSION PLAN COBERTURA** | data-object |  |  |
| **DAT01.EXCLUSION PLAN COBERTURA DETALLE** | data-object |  |  |
| **DAT01.FAC CONCEPTO PLAN** | data-object |  |  |
| **DAT01.FONDO** | data-object |  |  |
| **DAT01.FORMULA PLAN** | data-object |  |  |
| **DAT01.GENERO REQUISITO** | data-object |  |  |
| **DAT01.LIQUIDACION** | data-object |  |  |
| **DAT01.LIQUIDACION DETALLE** | data-object |  |  |
| **DAT01.NIVEL RIESGO COBERTURA** | data-object |  |  |
| **DAT01.NIVEL RIESGO CONFIG MOV** | data-object |  |  |
| **DAT01.NIVELES NOTAS PLAN COBERTURA** | data-object |  |  |
| **DAT01.NOTA ACLARATORIA PLAN** | data-object |  |  |
| **DAT01.NOTIFICACION EVENTO** | data-object |  |  |
| **DAT01.ORIGEN COBERTURA COBERTURA** | data-object |  |  |
| **DAT01.PERIODO CARENCIA** | data-object |  |  |
| **DAT01.PERSONA** | data-object |  |  |
| **DAT01.PLAN** | data-object |  |  |
| **DAT01.PLAN CANAL VENTA** | data-object |  |  |
| **DAT01.PLAN COBERTURA** | data-object |  |  |
| **DAT01.PLAN COBERTURA DEPENDIENTE** | data-object |  |  |
| **DAT01.PLAN COBERTURA EDAD** | data-object |  |  |
| **DAT01.PLAN COBERTURA REQUISITO** | data-object |  |  |
| **DAT01.PLAN FRECUENCIA FACTURACION** | data-object |  |  |
| **DAT01.PLAN MEDIO FACTURACION** | data-object |  |  |
| **DAT01.PLAN NIVEL RIESGO** | data-object |  |  |
| **DAT01.PLAN OBLIGATORIO** | data-object |  |  |
| **DAT01.PLAN PARENTESCO** | data-object |  |  |
| **DAT01.PLAN PERSEVERANTE** | data-object |  |  |
| **DAT01.PORCENTAJE CUOTA** | data-object |  |  |
| **DAT01.PRODUCTO COBERTURA** | data-object |  |  |
| **DAT01.PRODUCTO EXCLUYENTE** | data-object |  |  |
| **DAT01.PROMOCION CANAL** | data-object |  |  |
| **DAT01.PROMOCION CONDICION** | data-object |  |  |
| **DAT01.PROMOCION PLAN COBERTURA** | data-object |  |  |
| **DAT01.PROMOTOR CANAL** | data-object |  |  |
| **DAT01.RECONOCIMIENTO POR PERMANENCIA** | data-object |  |  |
| **DAT01.REGLAS EXCEPCIONES** | data-object |  |  |
| **DAT01.REQUISITO CONTROL MEDICO** | data-object |  |  |
| **DAT01.RESPONSABLE PERSONA** | data-object |  |  |
| **DAT01.SOLICITUD EVENT** | data-object |  |  |
| **DAT01.SOLICITUD EVENTO DETALLE** | data-object |  |  |
| **DAT01.SUBLIMITE COBERTURA** | data-object |  |  |
| **DAT01.SUBSISTENTE PLAN COBERTURA** | data-object |  |  |
| **DAT01.SUBSISTENTE PLAN COBERTURA DETALLE** | data-object |  |  |
| **DAT01.TIPO MOVIMIENTO** | data-object |  |  |
| **DAT01.TIPO PROCESO CUMULO** | data-object |  |  |
| **DAT01.TRANSACCION EXCLUSION** | data-object |  |  |
| **DAT01.TRANSACCION EXCLUSION COBERTURA** | data-object |  |  |
| **DAT01.TRANSACCION REQUISITO** | data-object |  |  |
| **DAT01.VALOR ASEGURADO** | data-object |  |  |
| **DAT01.VALOR ASEGURADO PLAN COBERTURA** | data-object |  |  |
| **DAT01.VALOR ASEGURADO TOPE** | data-object |  |  |
| **DAT01.VALOR CUOTA PLAN COBERTURA** | data-object |  |  |
| **DAT01.VALOR RESCATE** | data-object |  |  |

## MiMutual. 7b. Datos. Relaciones

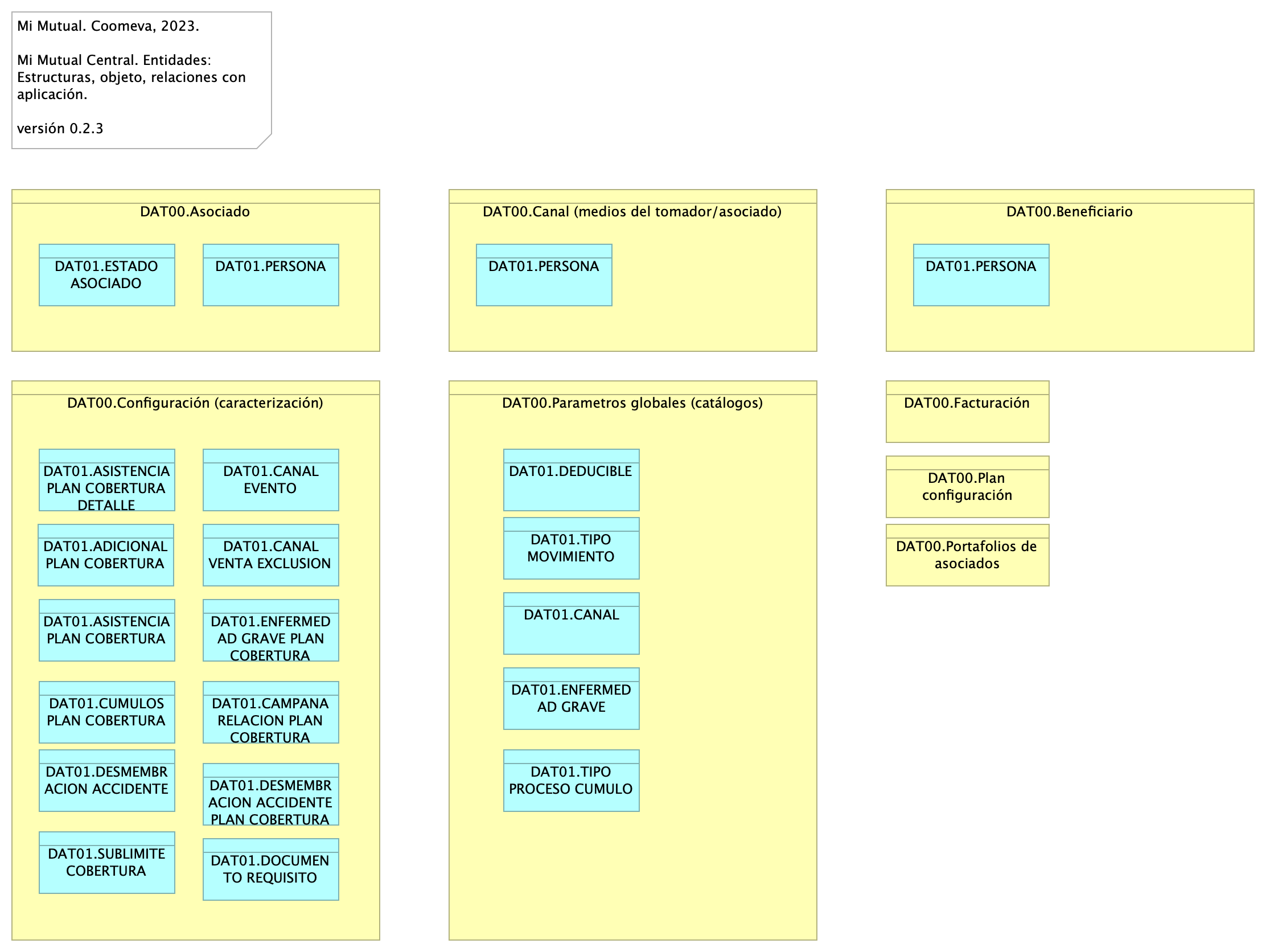


Figure 10: Diagram: MiMutual. 7b. Datos. Relaciones

### Entidades de Negocio Mi Mutual

* Configuración (caracterización de productos, plan)
* Plan (producto pólizas seguros)
* Canal (medios del tomador/asociado)
* Parametros globales (catálogos)
* Portafolio de asociado
* Asociado
* Facturación
* Beneficiario

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **DAT00.Asociado** | business-object |  |  |
| **DAT00.Beneficiario** | business-object |  |  |
| **DAT00.Canal (medios del tomador/asociado)** | business-object |  |  |
| **DAT00.Configuración (caracterización)** | business-object | Caracterización de productos, planes, parámetros |  |
| **DAT00.Facturación** | business-object | Factura la genera COOMEVA. |  |
| **DAT00.Parametros globales (catálogos)** | business-object |  |  |
| **DAT00.Plan configuración** | business-object | Plan de configuración: producto pólizas seguros. |  |
| **DAT00.Portafolios de asociados** | business-object |  |  |
| **DAT01.ADICIONAL PLAN COBERTURA** | data-object |  |  |
| **DAT01.ASISTENCIA PLAN COBERTURA** | data-object |  |  |
| **DAT01.ASISTENCIA PLAN COBERTURA DETALLE** | data-object |  |  |
| **DAT01.CAMPANA RELACION PLAN COBERTURA** | data-object |  |  |
| **DAT01.CANAL** | data-object |  |  |
| **DAT01.CANAL EVENTO** | data-object |  |  |
| **DAT01.CANAL VENTA EXCLUSION** | data-object |  |  |
| **DAT01.CUMULOS PLAN COBERTURA** | data-object |  |  |
| **DAT01.DEDUCIBLE** | data-object |  |  |
| **DAT01.DESMEMBRACION ACCIDENTE** | data-object |  |  |
| **DAT01.DESMEMBRACION ACCIDENTE PLAN COBERTURA** | data-object |  |  |
| **DAT01.DOCUMENTO REQUISITO** | data-object |  |  |
| **DAT01.ENFERMEDAD GRAVE** | data-object |  |  |
| **DAT01.ENFERMEDAD GRAVE PLAN COBERTURA** | data-object |  |  |
| **DAT01.ESTADO ASOCIADO** | data-object |  |  |
| **DAT01.PERSONA** | data-object |  |  |
| **DAT01.PERSONA** | data-object |  |  |
| **DAT01.PERSONA** | data-object |  |  |
| **DAT01.SUBLIMITE COBERTURA** | data-object |  |  |
| **DAT01.TIPO MOVIMIENTO** | data-object |  |  |
| **DAT01.TIPO PROCESO CUMULO** | data-object |  |  |

# Cotizador Web

## Cotizador. 1. Contexto



Figure 11: Diagram: Cotizador. 1. Contexto

### Contexto Mi Mutual Web

La aplicación Cotizador Web hace parte de los módulos de interfaz web de Mi Mutual Central, representado por API Mi Mutual en el diagrama. Realizar cotizaciones de los planes de protección luego de la vinculación del asociado.

La estructura por módulos permite realizar aplicaciones escalables y robustas ya que permite organizar las partes de la aplicación, la organización en bloques, extender la aplicación con funcionalidades de libreras externas, proporcionar un entorno de resolución de plantillas y además permite especificar la forma de la carga de cada uno de los componentes y servicios que conforman un módulo.

### Módulos Externos

Los módulos externos son todas y cada una de las herramientas que se utilizan para complementar con funcionalidades ya desarrolladas y tomadas desde un repositorio externo (NPM).

* TranslateModule: Manejo de internacionalización. Documentación: https://github.com/ngx-translate/core
* NgxMaskModule: Manejo de máscaras de input text. Documentación: https://github.com/JsDaddy/ngx-mask
* JwtModule: Manejo de token. Documentación: https://github.com/auth0/angular2-jwt
* sweetalert2: Manejo de alertas de mensajes. Documentación: https://sweetalert2.github.io/
* ngx-ui-loader: Manejo de Spinner para control de peticiones asíncronas. Documentación: https://github.com/t-ho/ngx-ui-loader
* Ngprime: Manejo de componentes visuales Documentación: https://www.primefaces.org/primeng/#/
* chart.js: componente utilizado para el manejo de graficas Documentación: https://www.chartjs.org/docs/latest/
* classlist.js: componete para el manejo de listas de datos en las gráficas Documentación: https://www.chartjs.org/docs/latest/
* cronstrue: componente para traducir una expresión cron a palabras Documentación: https://github.com/bradymholt/cronstrue
* file-saver: componente para descargar un archivo desde los bytes Documentación: https://github.com/eligrey/FileSaver.js#readme
* ngx-tinymce: Editor html para generación de plantillas para cartas Documentación: https://cipchk.github.io/ngx-tinymce/#/
* quill: componente para editor html Documentación: https://quilljs.com/

### Servicios Transversales

* AuthGuard: Validación de existencia de autenticación
* DeaciveGuard: Validación de salida de un componente
* ErrorInterceptor: Interceptor de Errores del back
* JwtInterceptor: Interceptor para inyectar el token
* AutenticationService: Métodos para completar la autenticación
* TypesService: Consumo de servicios de parametrización
* IdleTimeoutService: Verificación de timeout del token

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **Controlador Mi Mutual** | application-component | Los componentes de este tipo se encargan de controlar los servicios rest de la aplicación, además en estos componentes se define la forma como se reciben y envían los datos de los servicios rest y la seguridad de cada uno de los métodos. | *modulo:* mimutual |
| **app: Cotizador Web** | application-component | pkg: MiMutualWeb | *modulo:* cotizador |
| **(web) Cotizador** | application-function | Grupo de páginas web del cotizador. | *modulo:* cotizador |
| **(web) Proveedores** | application-function | Grupo de páginas web del cotizador. | *modulo:* cotizador |
| **(web) Reportes** | application-function | Grupo de páginas web del cotizador. | *modulo:* cotizador |
| **(web) admin Páginas** | application-function | Grupo de páginas web del cotizador. | *modulo:* cotizador |
| **Aplicativo** | application-function | Grupo de funcionalidades y entidades (datos) específicas del Cotizador Web. | *modulo:* cotizador |
| **Asociados** | application-function | Grupo de funcionalidad (servicios) de Asociados del Cotizador Web. | *modulo:* cotizador |
| **Cliente** | application-function | Grupo de funcionalidad (servicios) de Asociados del Cotizador Web. | *modulo:* cotizador |
| **Configuracn.** | application-function | Grupo de funcionalidad (servicios) de Planes de Configuración del Cotizador Web. | *modulo:* cotizador |
| **Cotizaciones** | application-function | Grupo de funcionalidad (servicios) de Cotización de la aplicación. | *modulo:* cotizador |
| **Interfaz gráfica** | application-function | Módulo interno (carpeta de proyecto) contenedor de las plantiilas de páginas web del Cotizador. | *modulo:* cotizador |
| **Módulos Compartidos** | application-function | Librerías de software base que el Cotizador Web requiere. Dependencias a paquetes de software de base, distintas a los módulos de negocio, necesarios para la ejecución de tareas utilitarias del Cotizador, tales como comunicación, políticas de seguridad, especificación de objetos globales de interfaz, transporte, transformación, entre otras. | *modulo:* cotizador |
| **Util** | application-function | En la Utilidades se especifican las clases que complementan una funcionalidad de un componente o servicio.\* FormValidate: Clase que implementa un disparador de validación de todos los campos de un formulario.\* CustomValidators: Creación de validaciones de campos. | *modulo:* cotizador |
| **admin Servicios** | application-function |  | *modulo:* cotizador |
| **API Mi Mutual** | application-interface |  | *modulo:* mimutual |
| **Autenticación: authgard** | application-service |  |  |
| **Interceptor: errorinterceptor** | application-service |  |  |
| **Parametrización: typeservice** | application-service |  |  |
| **Sesión admin: idletimeout** | application-service |  |  |

## Cotizador. 2. Contenedores



Figure 12: Diagram: Cotizador. 2. Contenedores

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **Repositorio Mi Mutual** | application-component | Antes SIPAS, Mi Mutual es una aplicación web compuesta por distintos módulos de software con arreglo a todas las actividades necesarias que soportan la operación de los productos y servicios que ofrece la Unidad de Solidaridad y Seguros de la Cooperativa.Para el manejo de la persistencia de datos se utilizará Spring Data el cual se apoya en la especificación de JPA y en la implementación de HIBERNATE además de complementar esta capa de persistencia con nuevas funcionalidades que facilitan el acceso a datos. | *modulo:* mimutual |
| **app: Cotizador Web** | application-component | pkg: MiMutualWeb | *modulo:* cotizador |
| **Control** | application-function |  |  |
| **Ruteos** | application-function |  |  |
| **Servicios** | application-function |  |  |
| **API Mi Mutual** | application-interface |  | *modulo:* mimutual |
| **Cliente HTTP / HTTPS** | application-interface |  |  |

## Cotizador. 4. Aplicación



Figure 13: Diagram: Cotizador. 4. Aplicación

La organización de la aplicación Cotizador Web Mi Mutual, como capa de presentación y servicios, plantea una estructura basada en la referencia de aplicaciones Angular 12. Las características de esta estructura (referida por Angular) está orientada al crecimiento (tamaño) de la aplicación, la escalabilidad y al rendimiento. La aplicación web Cotizador está diseñada (modulos) para manejar la carga por demanda del contenido.

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **Controlador Mi Mutual** | application-component | Los componentes de este tipo se encargan de controlar los servicios rest de la aplicación, además en estos componentes se define la forma como se reciben y envían los datos de los servicios rest y la seguridad de cada uno de los métodos. | *modulo:* mimutual |
| **MOD0.JwtModule** | application-component | Manejo de token. Documentación: https://github.com/auth0/angular2-jwt |  |
| **MOD0.Ngprime** | application-component | Manejo de componentes visuales Documentación: https://www.primefaces.org/primeng/#/ |  |
| **MOD0.NgxMaskModule** | application-component | Manejo de máscaras de input text. Documentación: https://github.com/JsDaddy/ngx-mask |  |
| **MOD0.TranslateModule** | application-component | Manejo de internacionalización. Documentación: https://github.com/ngx-translate/core |  |
| **MOD0.chart.js** | application-component | Componente utilizado para el manejo de graficas Documentación: https://www.chartjs.org/docs/latest/ |  |
| **MOD0.classlist.js** | application-component | Componete para el manejo de listas de datos en las gráficas Documentación: https://www.chartjs.org/docs/latest/ |  |
| **MOD0.cronstrue** | application-component | Componente para traducir una expresión cron a palabras Documentación: https://github.com/bradymholt/cronstrue |  |
| **MOD0.file-saver** | application-component | Componente para descargar un archivo desde los bytes Documentación: https://github.com/eligrey/FileSaver.js#readme |  |
| **MOD0.ngx-tinymce** | application-component | Editor html para generación de plantillas para cartas Documentación: https://cipchk.github.io/ngx-tinymce/#/ |  |
| **MOD0.ngx-ui-loader** | application-component | Manejo de Spinner para control de peticiones asíncronas. Documentación: https://github.com/t-ho/ngx-ui-loader |  |
| **MOD0.quill** | application-component | Ccomponente para editor html Documentación: https://quilljs.com/ |  |
| **MOD0.sweetalert2** | application-component | Manejo de alertas de mensajes. Documentación: https://sweetalert2.github.io/ |  |
| **Repositorio Mi Mutual** | application-component | Antes SIPAS, Mi Mutual es una aplicación web compuesta por distintos módulos de software con arreglo a todas las actividades necesarias que soportan la operación de los productos y servicios que ofrece la Unidad de Solidaridad y Seguros de la Cooperativa.Para el manejo de la persistencia de datos se utilizará Spring Data el cual se apoya en la especificación de JPA y en la implementación de HIBERNATE además de complementar esta capa de persistencia con nuevas funcionalidades que facilitan el acceso a datos. | *modulo:* mimutual |
| **app: Cotizador Web** | application-component | pkg: MiMutualWeb | *modulo:* cotizador |
| **app: Implementación de Servicios** | application-component | Los componentes de este tipo se encargan de controlar y almacenar toda la lógica del negocio, validaciones y todo lo referente a procesamiento de datos. | *modulo:* mimutual |
| **app: Mi Mutual Central** | application-component | Antes SIPAS, Mi Mutual es una aplicación web compuesta por distintos módulos de software con arreglo a todas las actividades necesarias que soportan la operación de los productos y servicios que ofrece la Unidad de Solidaridad y Seguros de la Cooperativa. | *modulo:* mimutual |
| **Interfaz transporte** | application-interface | Feign Client.Integración con otros sistemas para facilitar los procesos de vinculación, retiro, reactivación o fallecimiento de asociados. |  |
| **Administración** | application-service | Servicios de aplicación central que el Cotizador Web usa. Administración. |  |
| **Application Service** | application-service | Otros servicios del contexto de Mi Mutual Central. |  |
| **Cliente** | application-service | Servicios de aplicación central que el Cotizador Web usa. Operaciones sobre Clientes. |  |
| **Configuración** | application-service | Configuración o parametrización de factores para realizar los cálculos de las contribuciones de los asociados a la Cooperativa para cada uno de los productos adquiridos. |  |
| **Gestión de Productos** | application-service | Gestión de productos del fondo mutual y auxilio funerario que involucran lo relacionado a las siguientes coberturas: \* Fondo de Solidaridad: Incapacidades temporales, Incapacidades Permanentes (total, parcial), Perseverancia 60, 62, 65, 70 años, Perseverancias Anticipadas, Fallecimiento Asociado (Auxilio por muerte), Desempleo, Disminución de ingresos y enfermedades graves; Rentas por hospitalización, Enfermedades de Alto Costo, Pólizas de seguros personales y patrimoniales, Planes educativos, Segunda opinión médica, Asistencias. \* Auxilio Funerario: Fallecimiento de familiares directos (inscritos) del Asociado. |  |
| **Gestión de Usuarios** | application-service | Gestión de Usuarios: Administración de la información relacionada con los usuarios del sistema. Este componente se comunica con el servicio unificado de autenticación y autorización que devuelve los permisos que un usuario posee sobre las opciones que proporciona el sistema. |  |
| **Multiactiva** | application-service | Servicios de aplicación central que el Cotizador Web usa. Multiactiva. |  |
| **SS02.protecciones - mim - actuaria** | application-service | Servicios de aplicación central que el Cotizador Web usa. Protección. |  |
| **SS02.protecciones- desmemebracion - accidente** | application-service | Servicios de aplicación central que el Cotizador Web usa. Protección. |  |
| **SS02.reporte - cotizacion** | application-service | Servicios de aplicación central que el Cotizador Web usa. Reportes. |  |
| **SS02.reporte - estado - cotizacion** | application-service | Servicios de aplicación central que el Cotizador Web usa. Cotización. |  |
| **Simuladores** | application-service | Simuladores: Funcionalidades que permiten generar las simulaciones de los diferentes planes o modificaciones (incrementos y disminuciones) a los productos del Asociado. |  |
| **Analistas** | business-role | Analistas y auxiliares de servicio regional y nacional, agentes del centro de contacto, auditores médicos, analistas de operaciones (aseguramiento y facturación) y jefes. |  |
| **Asesores** | business-role | Asesores integrales |  |
| **Auxiliares servicio** | business-role | Analistas y auxiliares de servicio regional y nacional, agentes del centro de contacto, auditores médicos, analistas de operaciones (aseguramiento y facturación) y jefes. |  |
| **Módulos Externos** | grouping |  |  |
| **Unidad de Solidaridad y Seguros** | grouping | La Unidad de Solidaridad y Seguros cuenta con un software integrado para su core de negocio denominado SIPAS (Sistema de Previsión, Asistencia y Solidaridad) |  |
| **Servicio de Almacenamiento de Datos** | technology-service |  |  |
| **Servicio de Red** | technology-service |  |  |
| **Servicio de archivos** | technology-service |  |  |

## Cotizador. 4a. Aplicación. Servicios



Figure 14: Diagram: Cotizador. 4a. Aplicación. Servicios

Composición interna de los servivios de Mi Mutual Central, Mi Mutual Web, Cotizador Web.

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **Controlador** | application-component | Controlador interno del servicio. Punto de entrada a la lógica de expuesta. |  |
| **Interfaz** | application-component | Interfaz de inversión de dependencia a las clases del servicio. |  |
| **Interfaz datos** | application-component | Acceso a datos del modelo del contexto de Mi Mutual Central. |  |
| **Operación** | application-component |  |  |
| **Servicio** | application-component | Exposición de componentes de negocio. |  |
| **Interfaz transporte** | application-interface | Feign Client.Integración con otros sistemas para facilitar los procesos de vinculación, retiro, reactivación o fallecimiento de asociados. |  |
| **Application Service** | application-service | Otros servicios del contexto de Mi Mutual Central. |  |
| **Gestión de Productos** | application-service | Gestión de productos del fondo mutual y auxilio funerario que involucran lo relacionado a las siguientes coberturas: \* Fondo de Solidaridad: Incapacidades temporales, Incapacidades Permanentes (total, parcial), Perseverancia 60, 62, 65, 70 años, Perseverancias Anticipadas, Fallecimiento Asociado (Auxilio por muerte), Desempleo, Disminución de ingresos y enfermedades graves; Rentas por hospitalización, Enfermedades de Alto Costo, Pólizas de seguros personales y patrimoniales, Planes educativos, Segunda opinión médica, Asistencias. \* Auxilio Funerario: Fallecimiento de familiares directos (inscritos) del Asociado. |  |
| **Integración** | application-service |  |  |

## Cotizador. 4a. Dependencias



Figure 15: Diagram: Cotizador. 4a. Dependencias

### Paquetes y Dependencias Cotizador Web

Módulos y componentes que hacen parte de la estructura de la aplicación Cotizador Web (basado en Angular 12 [[1]](#footnote-118)).

### Módulos Cotizador Web

La estructura por módulos actual apunta a la escalabilidad y mantenimiento del Cotizador en términos de: organizar las partes de la aplicación, organización los bloques, extender la aplicación con libreras externas, proporcionar un entorno de resolución de plantillas y además, distribuir las cargas de los componentes y servicios que usa la aplicación.

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **app: Cotizador Web** | application-component | pkg: MiMutualWeb | *modulo:* cotizador |
| **pkg: admin** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: administración** | application-component | admin controller: Almacenan todas las clases que constituyen los servicios REST de la administrción de la aplicación. | *modulo:* cotizador |
| **pkg: asociados** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: auth** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: cliente** | application-component | controladores web de cliente. Reúne las clases que constituyen el modelo de entrada/salida de la interfaz gráfica de Clientes. Distinto al paquete admin controlador Clientes (pkg: web.clientes). | *modulo:* cotizador |
| **pkg: clientes** | application-component | admin controller: Almacenan todas las clases que constituyen los servicios REST de la administrción de la aplicación. Disitnto al paquete web de Cliente (pkg: admin.cliente). | *modulo:* cotizador |
| **pkg: componentes** | application-component | controller: contiene las clases que constituyen los llamados a librerías compartidas de la aplicación. | *modulo:* cotizador |
| **pkg: config** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: cotizaciones** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: cotización** | application-component | admin controller: Almacenan todas las clases que constituyen los servicios REST de la administrción de la aplicación. | *modulo:* cotizador |
| **pkg: directivas** | application-component | controller: contiene las clases que constituyen los llamados a librerías compartidas de la aplicación. | *modulo:* cotizador |
| **pkg: home** | application-component | admin controller: Almacenan todas las clases que constituyen los servicios REST de la administrción de la aplicación. | *modulo:* cotizador |
| **pkg: interfaces** | application-component | controller: contiene las clases que constituyen los llamados a librerías compartidas de la aplicación. | *modulo:* cotizador |
| **pkg: modelos** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: multiactiva** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: protecciones** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: proveedores** | application-component | admin controller: Almacenan todas las clases que constituyen los servicios REST de la administrción de la aplicación. | *modulo:* cotizador |
| **pkg: reporte** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: reportes** | application-component | admin controller: Almacenan todas las clases que constituyen los servicios REST de la administrción de la aplicación. | *modulo:* cotizador |
| **pkg: transporte** | application-component | controller: contiene las clases que constituyen los llamados a librerías compartidas de la aplicación. | *modulo:* cotizador |
| **pkg: util** | application-component | controller: contiene las clases que constituyen los llamados a librerías compartidas de la aplicación. | *modulo:* cotizador |
| **pkg: utilidades** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **(web) Cotizador** | application-function | Grupo de páginas web del cotizador. | *modulo:* cotizador |
| **(web) admin Páginas** | application-function | Grupo de páginas web del cotizador. | *modulo:* cotizador |
| **Aplicativo** | application-function | Grupo de funcionalidades y entidades (datos) específicas del Cotizador Web. | *modulo:* cotizador |
| **Interfaz gráfica** | application-function | Módulo interno (carpeta de proyecto) contenedor de las plantiilas de páginas web del Cotizador. | *modulo:* cotizador |
| **Módulos Compartidos** | application-function | Librerías de software base que el Cotizador Web requiere. Dependencias a paquetes de software de base, distintas a los módulos de negocio, necesarios para la ejecución de tareas utilitarias del Cotizador, tales como comunicación, políticas de seguridad, especificación de objetos globales de interfaz, transporte, transformación, entre otras. | *modulo:* cotizador |
| **Util** | application-function | En la Utilidades se especifican las clases que complementan una funcionalidad de un componente o servicio.\* FormValidate: Clase que implementa un disparador de validación de todos los campos de un formulario.\* CustomValidators: Creación de validaciones de campos. | *modulo:* cotizador |
| **admin Servicios** | application-function |  | *modulo:* cotizador |

## upd



Figure 16: Diagram: upd

### Especificaciones de Despliegue Cotizador Web

Detalles de configuración del proyecto Mi Mutual en el espacio de trabajo local (2022).

#### Recursos Requeridos

* Git. Se debe instalar git para poder realizar la clonación de cada uno de los proyectos mas adelante.
* Instalación SmartGit. Se debe instalar Smartgit para poder realizar la clonación de cada uno de los proyectos mas adelante, este es opcional ya que es una interfaz gráfica de git mas amigable para el usuario en caso que no desee trabajar con la consola.
* DBeaver. Se debe instalar DBeaver para poder acceder a la base de datos.
* Instalación Maven. Se debe instalar maven para poder compilar los proyectos, nos debemos asegurar de instalar la versión 3.6.3, en caso que no se encuentra en la página oficial copiar la carpeta que esta en el repositorio a archivo de programas.
* Java 8. Se debe instalar Java para poder desplegar los proyectos mas adelante, nos debemos asegurar de instalar la versión 8.
* STS. Se debe instalar el IDE para realizar modificaciones a los proyectos back mas adelante en este caso Spring Tools 4 for Eclipse. La carpeta que genera el instalador la copiamos a archivos de programa.
* Instalación Lombok. Se debe instalar el lombok seleccionando el IDE que acabamos de instarlar en este caso el STS.
* Postman. Se debe instalar el postman para poder consumir los servicios del backend mas adelante cuando ya se hayan desplegado.
* Node Js. Se debe instalar Node Js para configurar el proyecto front mas adelante, nos debemos asegurar de instalar la versión v14.2.0.
* Visual Studio Code. Se debe instalar el IDE para realizar modificaciones al proyecto front mas adelante en este caso Visual Studio code.
* Angular 14.

Nota: los paquetes con el mismo nombre, como pkg: cliente, y pkg: clientes que aparecen arriba en el diagrama corresponden a espacio de nombres distintos. Por ejemplo, para el caso de estos dos paquetes, pkg: cliente pertenece el espacio de nombre (web) Cotizador; en cambio, pkg: clientes pertenece al espacio de nombres (web) admin Páginas.

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **app: Asociados** | application-component | Contiene todas las funcionalidades relacionadas con consulta y creación de asociados y beneficiarios. | *modulo:* mimutual |
| **app: Cotizador Web** | application-component | pkg: MiMutualWeb | *modulo:* cotizador |
| **app: Implementación de Servicios** | application-component | Los componentes de este tipo se encargan de controlar y almacenar toda la lógica del negocio, validaciones y todo lo referente a procesamiento de datos. | *modulo:* mimutual |
| **app: Protecciones** | application-component | Contiene todas las funcionalidades relacionadas con la gestión y configuración de productos y protecciones. | *modulo:* mimutual |
| **app: Reclamaciones** | application-component | Contiene todas las funcionalidades relacionadas con la gestión de reclamaciones, liquidaciones y pagos. | *modulo:* mimutual |
| **pkg: admin** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: administración** | application-component | admin controller: Almacenan todas las clases que constituyen los servicios REST de la administrción de la aplicación. | *modulo:* cotizador |
| **pkg: asociados** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: auth** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: cliente** | application-component | controladores web de cliente. Reúne las clases que constituyen el modelo de entrada/salida de la interfaz gráfica de Clientes. Distinto al paquete admin controlador Clientes (pkg: web.clientes). | *modulo:* cotizador |
| **pkg: clientes** | application-component | admin controller: Almacenan todas las clases que constituyen los servicios REST de la administrción de la aplicación. Disitnto al paquete web de Cliente (pkg: admin.cliente). | *modulo:* cotizador |
| **pkg: config** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: cotizaciones** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: cotización** | application-component | admin controller: Almacenan todas las clases que constituyen los servicios REST de la administrción de la aplicación. | *modulo:* cotizador |
| **pkg: home** | application-component | admin controller: Almacenan todas las clases que constituyen los servicios REST de la administrción de la aplicación. | *modulo:* cotizador |
| **pkg: modelos** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: multiactiva** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: protecciones** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: proveedores** | application-component | admin controller: Almacenan todas las clases que constituyen los servicios REST de la administrción de la aplicación. | *modulo:* cotizador |
| **pkg: reporte** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **pkg: reportes** | application-component | admin controller: Almacenan todas las clases que constituyen los servicios REST de la administrción de la aplicación. | *modulo:* cotizador |
| **pkg: utilidades** | application-component | controller: Almacenan todas las clases que constituyen los servicios rest de la aplicación. | *modulo:* cotizador |
| **Conexión: jdbc** | artifact |  | *modulo:* cotizador |
| **Entorno Angular: ng 14.0.0** | system-software |  | *modulo:* cotizador |
| **Entorno JS: node 14.2.0** | system-software |  | *modulo:* cotizador |
| **Repositorio: db2 iSerie** | system-software |  | *modulo:* cotizador |
| **Servicios: tomcat** | system-software |  | *modulo:* mimutual |

## Cotizador. 7. Datos. Negocio



Figure 17: Diagram: Cotizador. 7. Datos. Negocio

### Entidades de Negocio Mi Mutual

Dominios de datos de negocio. Entidades independiente de la plataforma y de la tecnología.

* Configuración (caracterización de productos, plan)
* Plan (producto pólizas seguros)
* Canal (medios del tomador/asociado)
* Parametros globales (catálogos)
* Portafolio de asociado
* Asociado
* Facturación
* Beneficiario

### Catálogo de Elementos

| Name | Type | Description | Properties |
| --- | --- | --- | --- |
| **DAT00. Cobertura** | business-object |  |  |
| **DAT00.Asegurado** | business-object |  |  |
| **DAT00.Asociado** | business-object |  |  |
| **DAT00.Beneficiario** | business-object |  |  |
| **DAT00.Canal (medios del tomador/asociado)** | business-object |  |  |
| **DAT00.Configuración (caracterización)** | business-object | Caracterización de productos, planes, parámetros |  |
| **DAT00.Cotización** | business-object |  |  |
| **DAT00.Facturación** | business-object | Factura la genera COOMEVA. |  |
| **DAT00.Plan - Cobertura** | business-object |  |  |
| **DAT00.Plan configuración** | business-object | Plan de configuración: producto pólizas seguros. |  |
| **DAT00.Plan de Pagos** | business-object |  |  |
| **DAT00.Planes** | business-object |  |  |
| **DAT00.Producto** | business-object |  |  |
| **DAT00.Venta** | business-object |  |  |
| **DAT00.Vinculación** | business-object |  |  |
| **DAT01.CANAL\_CONFIG\_MOV** | data-object |  |  |
| **DAT01.CANAL\_EVENTO** | data-object |  |  |
| **DAT01.CANAL\_VENTA\_EXCLUSION** | data-object |  |  |
| **DAT01.CANAL\_VENTA\_EXCLUSION\_COBERTURA** | data-object |  |  |
| **DAT01.CICLO\_FACTURACION** | data-object |  |  |
| **DAT01.COBERTURA** | data-object |  |  |
| **DAT01.COTIZACION\_APORTE\_ESTATUTARIO\_ASEGURADO** | data-object |  |  |
| **DAT01.COTIZACION\_ASEGURADO\_TEMP** | data-object |  |  |
| **DAT01.COTIZACION\_ASEGURADO\_TEMP** | data-object |  |  |
| **DAT01.COTIZACION\_DETALLE\_TEMP** | data-object |  |  |
| **DAT01.COTIZACION\_DETALLE\_TEMP** | data-object |  |  |
| **DAT01.COTIZACION\_PLAN\_TEMP** | data-object |  |  |
| **DAT01.COTIZACION\_PLAN\_TEMP** | data-object |  |  |
| **DAT01.COTIZACION\_TEMP** | data-object |  |  |
| **DAT01.COTIZACION\_TEMP** | data-object |  |  |
| **DAT01.ESTADO\_COTIZACION** | data-object |  |  |
| **DAT01.ESTADO\_VENTA** | data-object |  |  |
| **DAT01.FRECUENCIA\_FACTURACION** | data-object |  |  |
| **DAT01.FRECUENCIA\_FACTURACION** | data-object |  |  |
| **DAT01.MEDIO\_FACTURACION** | data-object |  |  |
| **DAT01.MEDIO\_FACTURACION** | data-object |  |  |
| **DAT01.MIM\_COTIZACION\_APORTE\_ESTATUTARIO\_ASEGURADO** | data-object |  |  |
| **DAT01.MOVIMIENTO\_PLAN\_CANAL** | data-object |  |  |
| **DAT01.PERSONA** | data-object |  |  |
| **DAT01.PERSONA** | data-object |  |  |
| **DAT01.PERSONA** | data-object |  |  |
| **DAT01.PERSONA** | data-object |  |  |
| **DAT01.PLAN** | data-object |  |  |
| **DAT01.PLAN\_CANAL\_VENTA** | data-object |  |  |
| **DAT01.PLAN\_CANAL\_VENTA** | data-object |  |  |
| **DAT01.PLAN\_COBERTURA** | data-object |  |  |
| **DAT01.PLAN\_COBERTURA\_DEPENDIENTE** | data-object |  |  |
| **DAT01.PLAN\_COBERTURA\_EDAD** | data-object |  |  |
| **DAT01.PLAN\_COBERTURA\_TIPO\_COBERTURA** | data-object |  |  |
| **DAT01.PLAN\_FRECUENCIA\_FACTURACION** | data-object |  |  |
| **DAT01.PLAN\_FRECUENCIA\_FACTURACION** | data-object |  |  |
| **DAT01.PLAN\_MEDIO\_FACTURACION** | data-object |  |  |
| **DAT01.PLAN\_MEDIO\_FACTURACION** | data-object |  |  |
| **DAT01.PLAN\_NIVEL\_RIESGO** | data-object |  |  |
| **DAT01.PLAN\_OBLIGATORIO** | data-object |  |  |
| **DAT01.PLAN\_PARENTESCO** | data-object |  |  |
| **DAT01.PLAN\_PERSEVERANTE** | data-object |  |  |
| **DAT01.PRE\_VENTA** | data-object |  |  |
| **DAT01.PRODUCTO\_COBERTURA** | data-object |  |  |
| **DAT01.PROMOTOR\_CANAL** | data-object |  |  |
| **DAT01.PROSPECTO\_ASOCIADO\_COTIZACION** | data-object |  |  |
| **DAT01.RESPONSABLE\_PERSONA** | data-object |  |  |
| **DAT01.RESPONSABLE\_PERSONA** | data-object |  |  |
| **DAT01.SIP\_PRODUCTOS** | data-object |  |  |
| **DAT01.SIP\_PRODUCTOS\_TIPO** | data-object |  |  |
| **DAT01.TIPO\_COTIZACION** | data-object |  |  |
| **DAT01.TIPO\_VENTA** | data-object |  |  |
| **DAT01.VENTA** | data-object |  |  |
| **DAT01.VENTAS\_PREGUNTAS** | data-object |  |  |
| **DAT01.VENTA\_ASEGURADO** | data-object |  |  |
| **DAT01.VENTA\_DETALLE** | data-object |  |  |
| **DAT01.VENTA\_PLAN** | data-object |  |  |
| **DAT10.COBERTURA\_BENEFICIARIO** | data-object |  |  |

Generated on: Tue Nov 07 2023 10:24:20 GMT-0500 (COT)

1. Angular 2 tiene una arquitectura Modelo Vista Controlador (MVC) con el fin de facilitar el desarrollo gestionado. [↑](#footnote-ref-118)