DEFINICIÓN Y DESARROLLO DEL MÓDULO GRÁFICO DE IDbox

(REDESIGNING IDbox's GRAPHIC MODULE)

Isaac Gutiérrez Rodríguez

Trabajo de Fin de Grado para acceder al
GRADO EN INGENIERÍA INFORMÁTICA

Director: Carlos Blanco Bueno
Co-Director: Miguel Sierra Sánchez

JUNIO - 2018

ÍNDICE

Contexto

Motivación TFG

Objetivos del Proyecto

Metodología

Presentación del Problema

Requisitos

Diseño Arquitectónico

Diseño Detallado

Pruebas

Demo

Conclusiones

Trabajos Futuros

CONTEXTO

Sistema SCADA

- Supervisión, Control y Adquisición de Datos.
- Procesos industriales
- Componentes:
  - Sensores y Actuadores
  - Controladores Lógicos Programables (PLC)
  - Interfaz Gráfica

IDbox

- Sistema SCADA.
- Supervisión mediante:
  - Gráficas (Agrupados, Predicción, Tendencia, Comparación, Correlación, Dispersión)
  - Sinópticos

ÍNDICE

Contexto
Motivación TFG
Objetivos del Proyecto
Metodología
Presentación del Problema
Requisitos
Diseño Arquitectónico
Diseño Detallado
Pruebas
Demo
Conclusiones
Trabajos Futuros

MOTIVACIÓN TFG

Actualmente IDbox:

- 6 Tipos de Gráficas:
  - ❖ Agrupados
  - ❖ Tendencia
  - ❖ Correlación
  - ❖ Predicción
  - ❖ Comparación
  - ❖ Dispersión
- Desarrollo y mantenimiento de cada tipo por separado

Resultado:

- Desarrollo y mantenimiento ineficiente
- Incremento de los tiempos de desarrollo y mantenimiento
- Incremento de recursos necesarios para el desarrollo y mantenimiento

ÍNDICE

Contexto
Motivación TFG
Objetivos del Proyecto
Metodología
Presentación del Problema
Requisitos
Diseño Arquitectónico
Diseño Detallado
Pruebas
Demo
Conclusiones
Trabajos Futuros

OBJETIVO GLOBAL

Creación de un nuevo modelo de gráficas que unifique todos los tipos de gráficas existentes; de forma que su gestión, mantenimiento y desarrollo resulte más eficiente.

OBJETIVOS DEL PROYECTO

- Definición del modelo de Base de Datos (BBDD).
- Diseño e implementación del acceso a BBDD para la recuperación y persistencia de las gráficas.
- Diseño e implementación de todo el negocio relacionado con las gráficas de IDbox.
- Diseño e implementación de los servicios que ofrecerán las distintas funcionalidades gráficas de IDbox.
- Diseño e implementación de una Web API para el consumo externo de gráficas.

ÍNDICE

Contexto
Motivación TFG
Objetivos del Proyecto
Metodología
Presentación del Problema
Requisitos
Diseño Arquitectónico
Diseño Detallado
Pruebas
Demo
Conclusiones
Trabajos Futuros

METODOLOGÍA

Metodología Iterativa e Incremental

- 3 Iteracciones principales + Iteracción Previa
  - **Fases Iteracción Previa:** Análisis de Requisitos + Diseño Arquitectónico
  - **Fases iteracciones Principales:** Diseño Detallado + Implementación + Pruebas + Documentación
- **Duración:** 6 meses - 600h approx
- Tecnologías:
  - ASP.NET, Web API, SQL Server, Entity Framework, Dapper.
- Herramientas:
  - Visual Studio, SQL Server Management Studio, SQL Server Profiler, Ninject, Moq, JMeter, OpenCover, Advanced Rest Client, Git, MagicDraw, Project.

METODOLOGÍA

Nombre de tarea	Duración
Definición del Módulo Gráfico de IDBox	**124 días**
Formación inicial	3 días
Análisis de Requisitos	3.5 días
Diseño Arquitectónico	2 días
Iteracción 1: Diseño del Modelo de BBDD y Capa de Acceso a Datos	**60.5 días**
Revisión Diseño Arquitectónico	0.5 días
Diseño Detallado	4 días
Implementación	**42 días**
Análisis de Alternativas	1 día
Formación en el uso de las Herramientas y Tecnologías	10 días
Implementación de las clases de dominio	2 días
Comparación de alternativas	15 días
Completada elección del Framework a utilizar para generar la capa de acceso a datos	0 días
Implementación mediante el uso del framework elegido	14 días
Pruebas	14 días
Documentación	2 días
Primera iteración completada	0 días
Iteracción 2: Capa de Negocio	**10.5 días**
Revisión Diseño Arquitectónico	0.5 días
Diseño Detallado	2 días
Implementación	3 días
Pruebas	5 días
Documentación	2 días
Segunda Iteracción completada	0 días
Iteracción 3: WebApi	**44.5 días**
Revisión Diseño Arquitectónico	0.5 días
Diseño Detallado	3 días
Implementación	18 días
Pruebas	16 días
Optimizaciones	12 días
Documentación	2 días
Tercera iteración completada	0 días

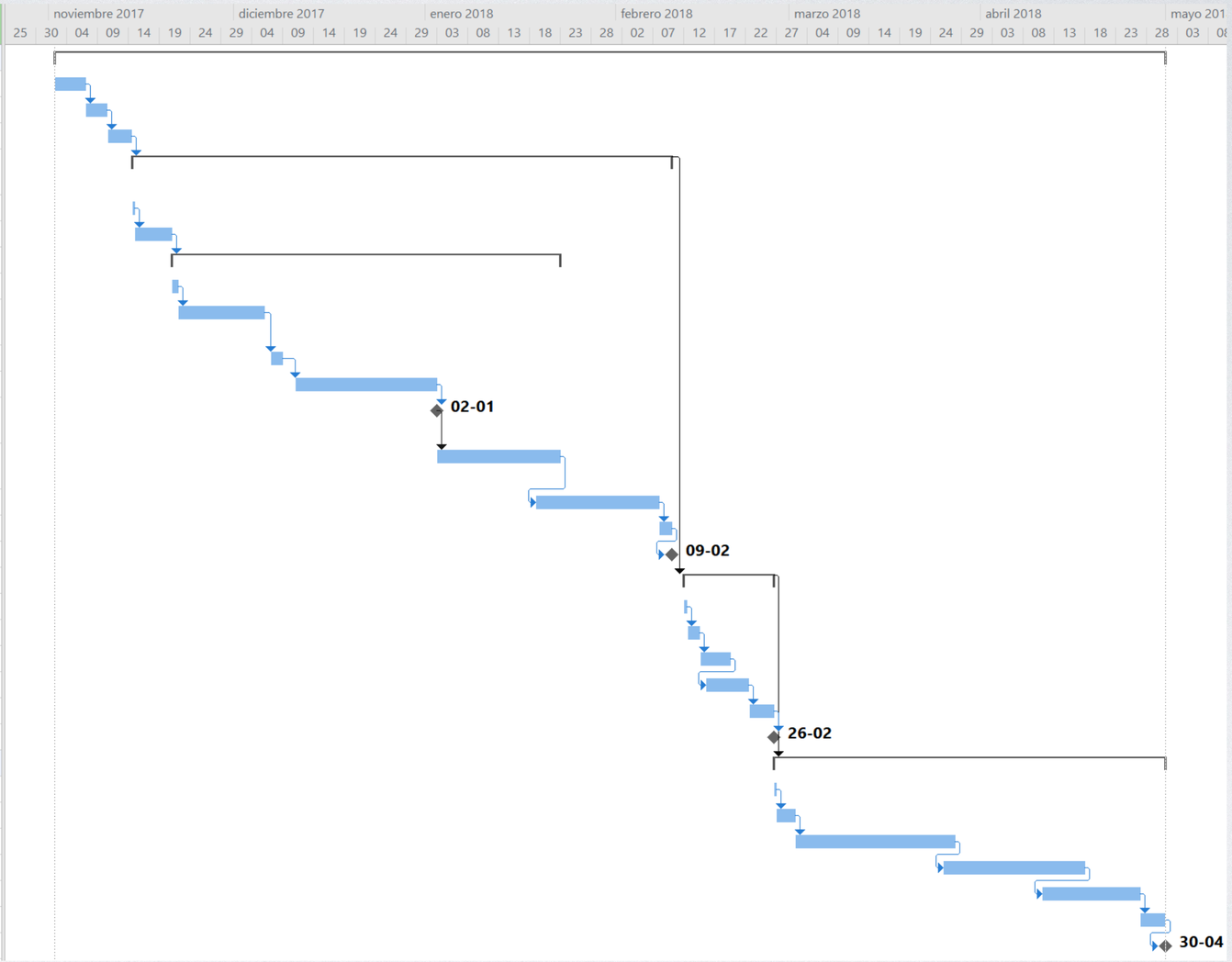


Diagrama Gantt del Plan de Trabajo

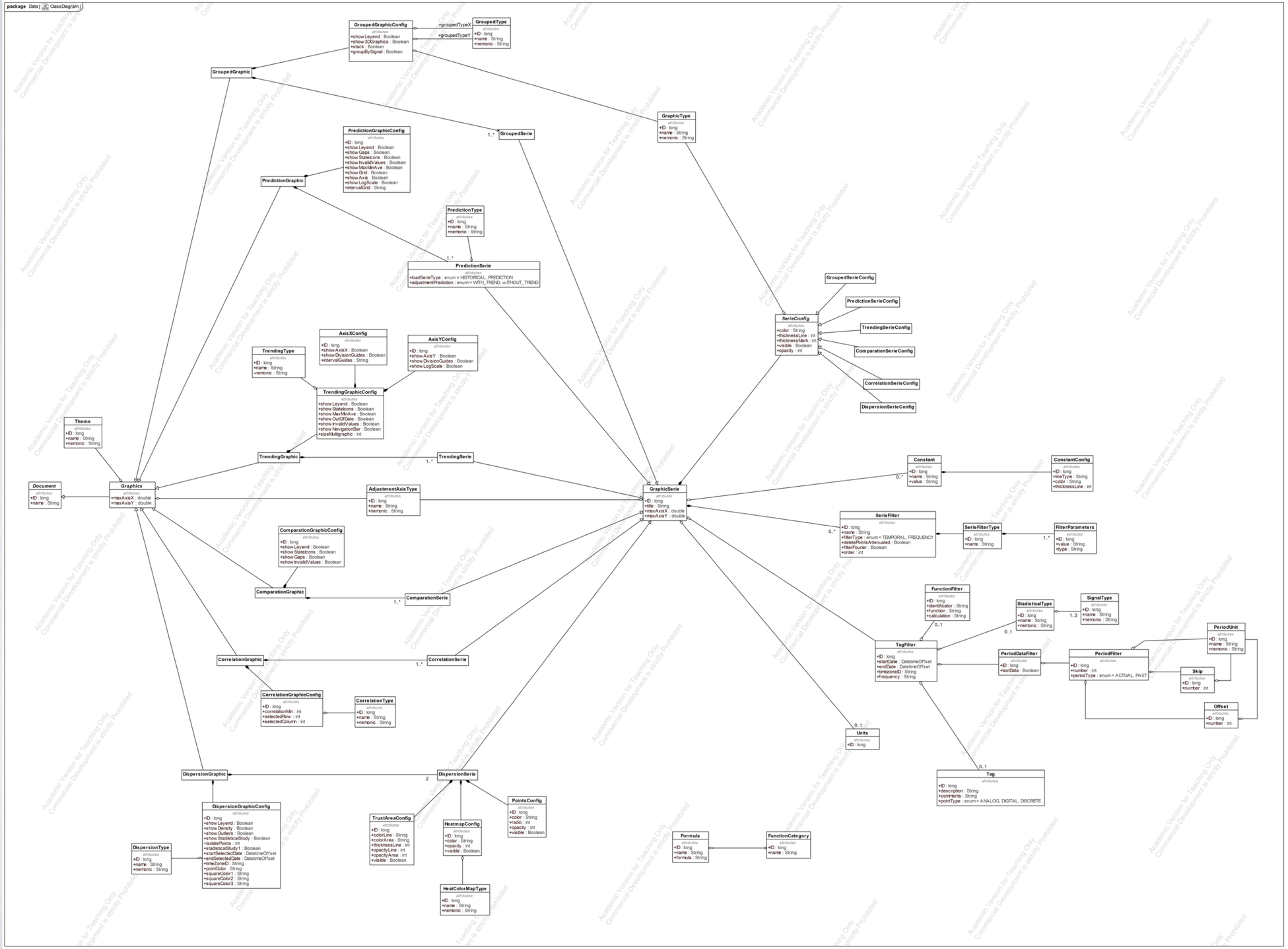
ÍNDICE

Contexto
Motivación TFG
Objetivos del Proyecto
Metodología
Presentación del Problema
Requisitos
Diseño Arquitectónico
Diseño Detallado
Pruebas
Demo
Conclusiones
Trabajos Futuros

PRESENTACIÓN DEL PROBLEMA

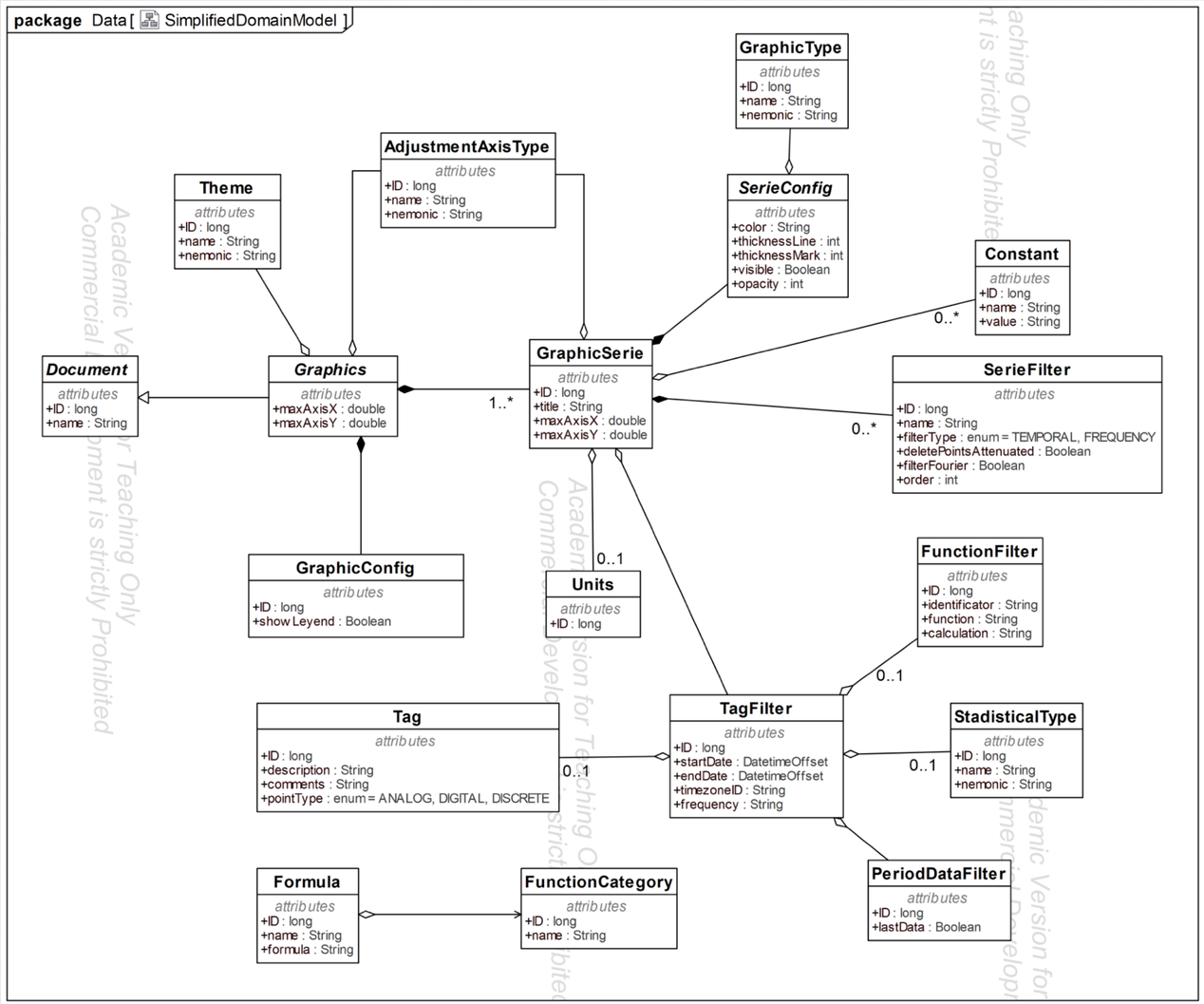
6 Tipos de Gráficas

- **Agrupados**
  - Agrupación de los valores de una señal en escalas temporales: Permite detectar patrones horarios, semanales,…
- **Predicción**
  - Visualización de datos históricos junto con datos futuros calculados en base a los históricos.
- **Tendencia**
  - Visualización de datos históricos de una señal.
- **Comparación**
  - Comparación de los valores de varios conjuntos de datos de una o varias señales.
- **Correlación**
  - Grado de correlación lineal existente entre dos señales.
- **Dispersión**
  - Relación existente entre dos señales, representando cada serie de datos en uno de los ejes.



Academic Version for Teaching Only
Commercial Development is strictly Prohibited

PRESENTACIÓN DEL PROBLEMA



Modelo de Dominio Simplificado

Gráfica:

- Visualización configurable
- Compuesta por Series
  - ❖ Asociadas a una señal o constante
  - ❖ Parámetros de Filtrado

ÍNDICE

Contexto
Motivación TFG
Objetivos del Proyecto
Metodología
Presentación del Problema
Requisitos
Diseño Arquitectónico
Diseño Detallado
Pruebas
Demo
Conclusiones
Trabajos Futuros

REQUISITOS FUNCIONALES

- ***Requisito Global:*** Dar soporte a la persistencia y recuperación de las entidades contenidas en el modelo de dominio anterior.

- ***Requisitos Concretos.*** Pueden dividirse en 4 puntos:
  - ❖ Gráficas
  - ❖ Tipos Preestablecidos
  - ❖ Fórmulas
  - ❖ Web API

REQUISITOS FUNCIONALES

Gráficas.

- El usuario ha de poder:
  - ***RF001:*** Recuperar una gráfica a través de su identificador.
  - ***RF002:*** Crear nuevas gráficas.
  - ***RF003:*** Modificar las propiedades de una gráfica.
  - ***RF004:*** Eliminar una gráfica.
  - ***RF005:*** Consultar el listado de gráficas.
  - ***RF006:*** Buscar gráficas en base a filtros preestablecidos.
- Otros:
  - ***RF007:*** Validar los datos de las gráficas de forma previa a realizar una operación que modifique el estado del sistema.
  - ***RF008:*** Gestión de la concurrencia: Control de cambios realizados.

REQUISITOS FUNCIONALES

Tipos Preestablecidos

- El usuario ha de poder consultar:
  - ***RF009:*** Los datos de un valor de un tipo preestablecido mediante su identificador o nemónico.
  - ***RF010:*** El listado de valores posibles para cada uno de los tipos preestablecidos.
  - ***RF011:*** El listado de tipos estadísticos compatibles con un tipo de señal.
  - ***RF012:*** Todas las señales de un determinado tipo de punto.

REQUISITOS FUNCIONALES

Fórmulas

- El usuario ha de poder:
  - ***RF013:*** Consultar los datos de una categoría de funciones mediante su identificador o nemónico.
  - ***RF014:*** Consultar el listado de categorías de funciones.
  - ***RF015:*** Consultar los datos de una fórmula mediante su identificador.
  - ***RF016:*** Crear nuevas fórmulas.
  - ***RF017:*** Modificar formulas.
  - ***RF018:*** Eliminar fórmulas.
  - ***RF019:*** Consultar el listado completo de fórmulas.
  - ***RF020:*** Consultar el listado de fórmulas pertenecientes a una categoría de funciones.
  - ***RF021:*** Gestión de la concurrencia: Control de cambios realizados.

REQUISITOS FUNCIONALES

Web API

- La Web API:
  - ***RF022:*** Soportará el intercambio de información mediante JSON y XML.
  - ***RF023:*** Dispondrá de una página de ayuda en la que se establecerán:
    - Los recursos disponibles.
    - La forma en que podrán ser accedidos (URI).
    - Los métodos HTTP soportados por cada recurso.
    - Los parámetros disponibles para cada método.

REQUISITOS NO FUNCIONALES

5 Categorias (ISO/IEC 25010):

- Rendimiento:
  - RNF001: Tiempos de respuesta por < 1s
  - RNF002: Acceso concurrente de 100s de usuarios manteniendo tiempos < 1s
- Mantenibilidad:
  - RNF004: Diseño modular para facilitar su mantenimiento y extensión
  - RNF005: Soporte de diferentes SGBDR mediante una única capa de acceso a datos
- Seguridad:
  - RNF003: Evitar Inyección SQL
- Usabilidad:
  - RNF006: Lengua inglesa en código y página de ayuda
- Accesibilidad:
  - RNF007: Web API accesible desde navegadores, aplicaciones de escritorio y aplicaciones móviles.

ÍNDICE

Contexto
Motivación TFG
Objetivos del Proyecto
Metodología
Presentación del Problema
Requisitos
Diseño Arquitectónico
Diseño Detallado
Pruebas
Demo
Conclusiones
Trabajos Futuros

ARQUITECTURA

Arquitectura en N-Capas:

- Persistencia
  - ❖ Recuperación y persistencia mediante Dapper.
- Negocio
  - ❖ Validación y conversión de datos.
- Controladores de la Web API
  - ❖ Definición de los recursos de la Web API y la forma de acceder a ellos.

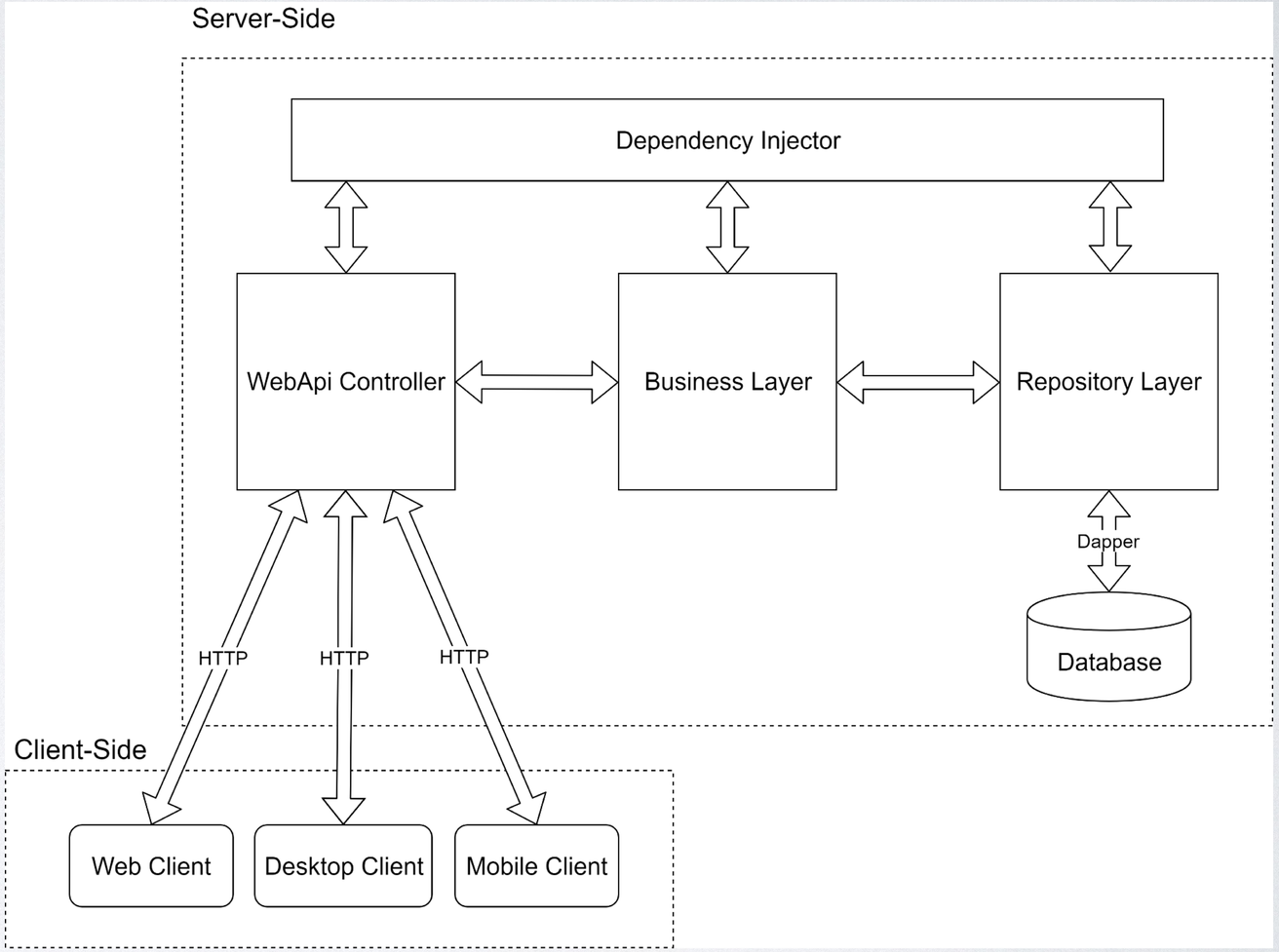
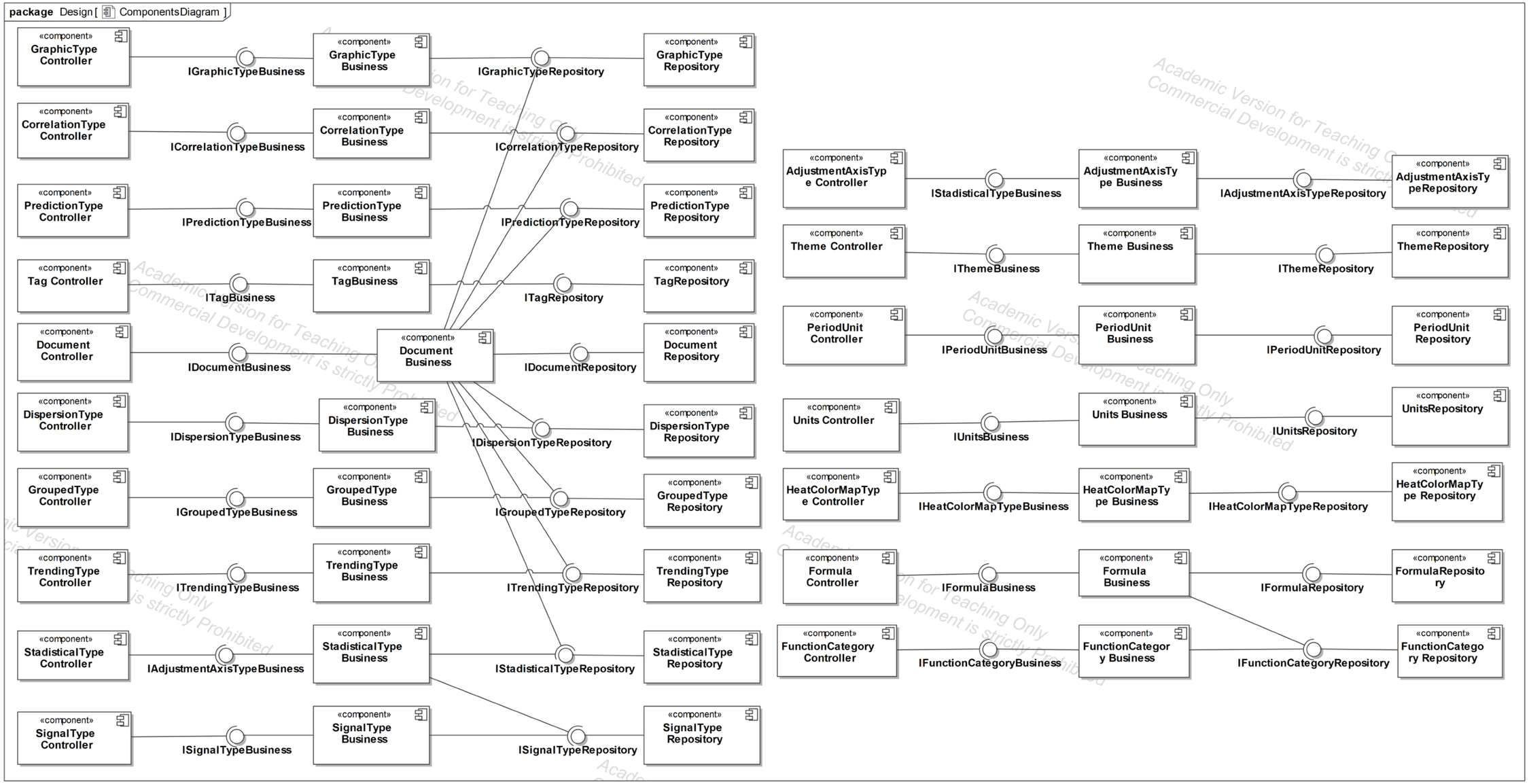
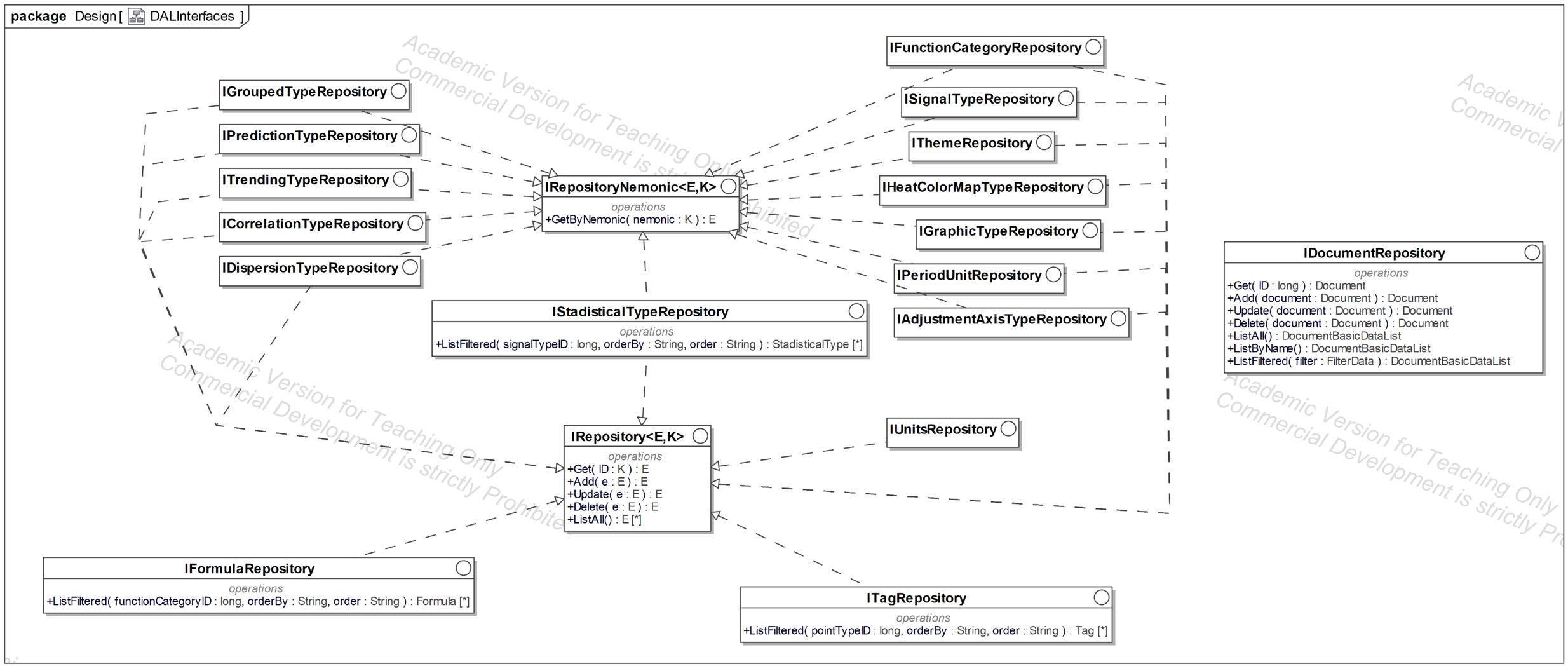


Diagrama de Arquitectura

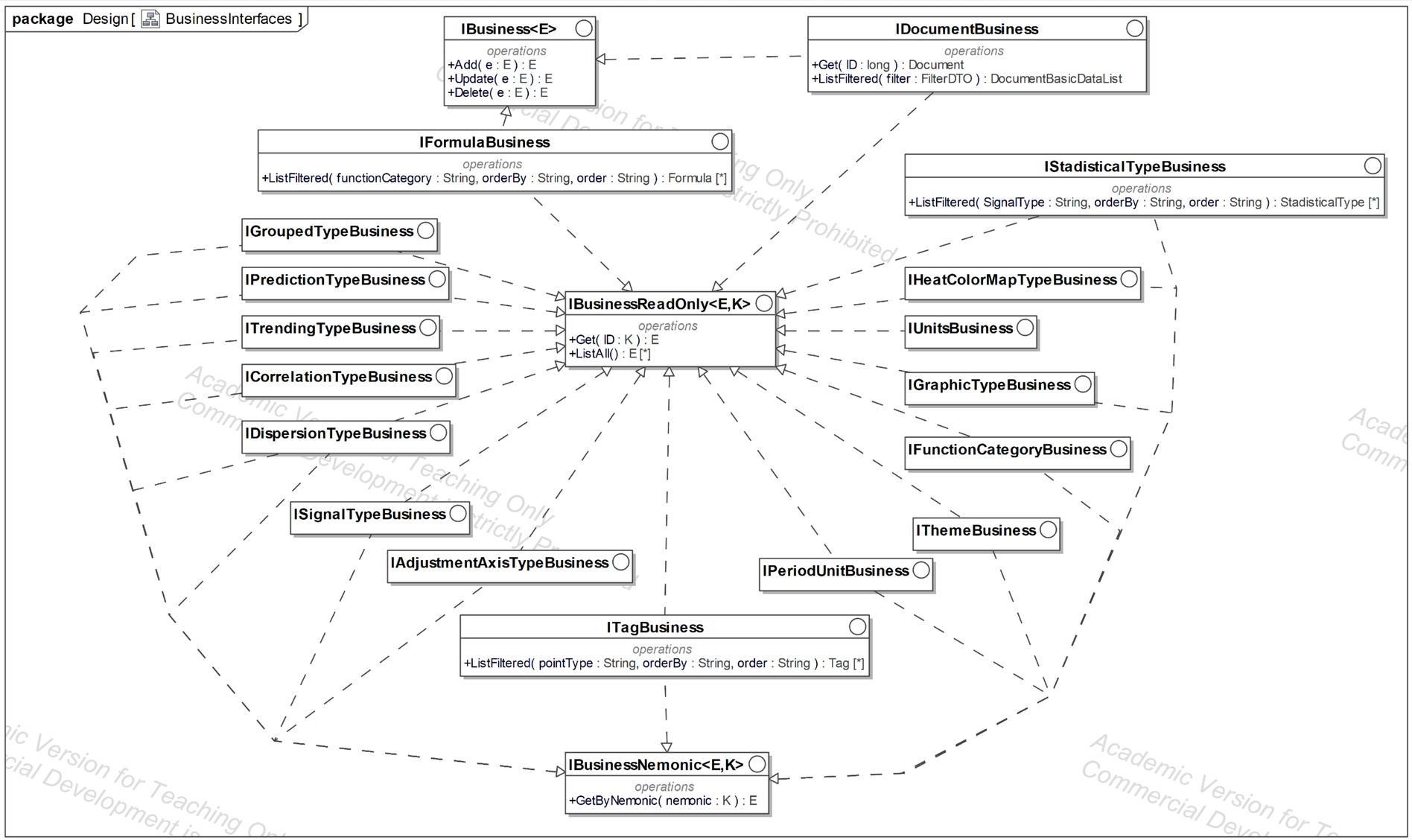
COMPONENTES



INTERFACES PERSISTENCIA



INTERFACES NEGOCIO



ÍNDICE

Contexto
Motivación TFG
Objetivos del Proyecto
Metodología
Presentación del Problema
Requisitos
Diseño Arquitectónico
Diseño Detallado
Pruebas
Demo
Conclusiones
Trabajos Futuros

DISEÑO DETALLADO PERSISTENCIA

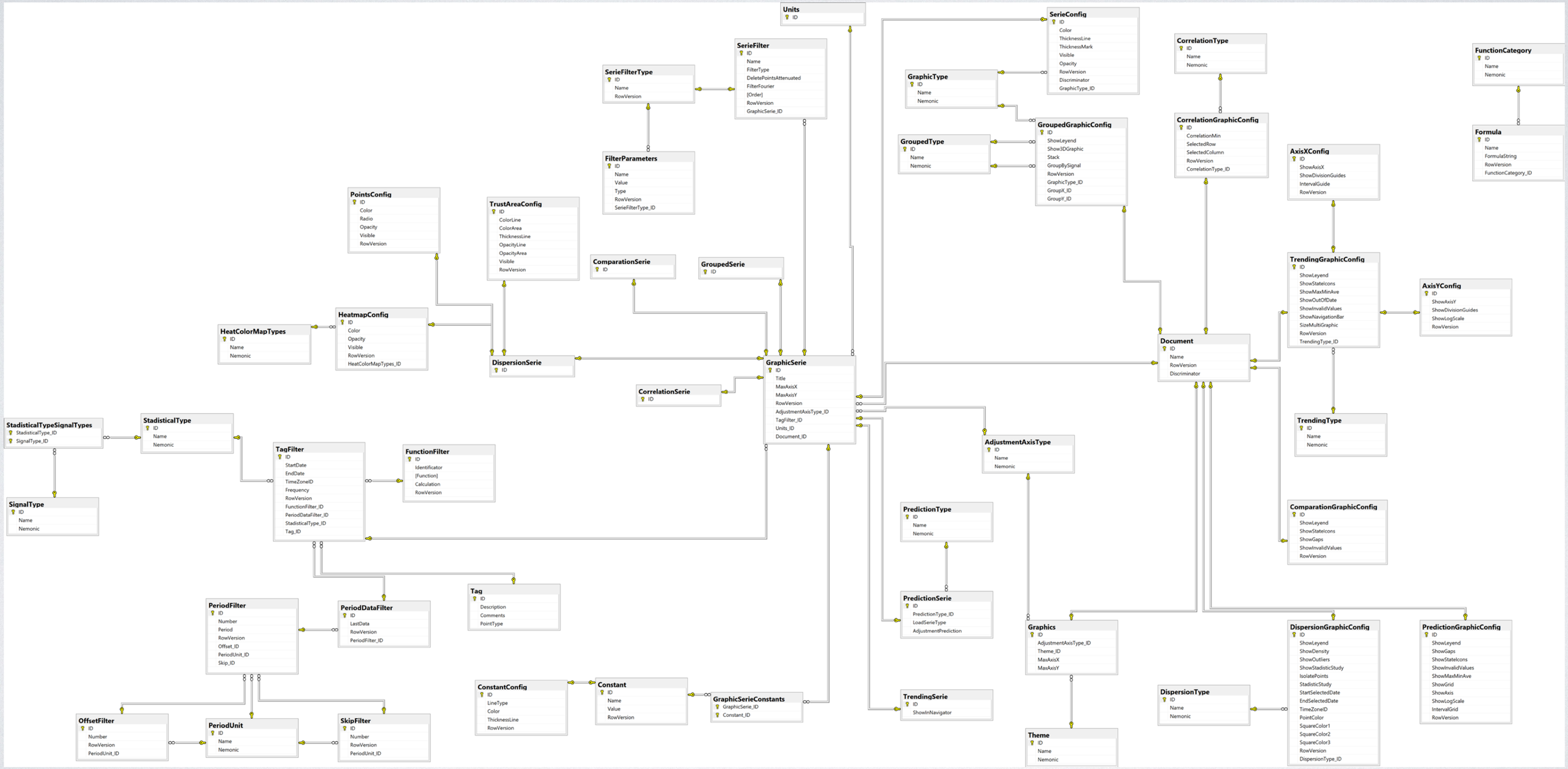


Diagrama SQL Completo

DISEÑO DETALLADO PERSISTENCIA

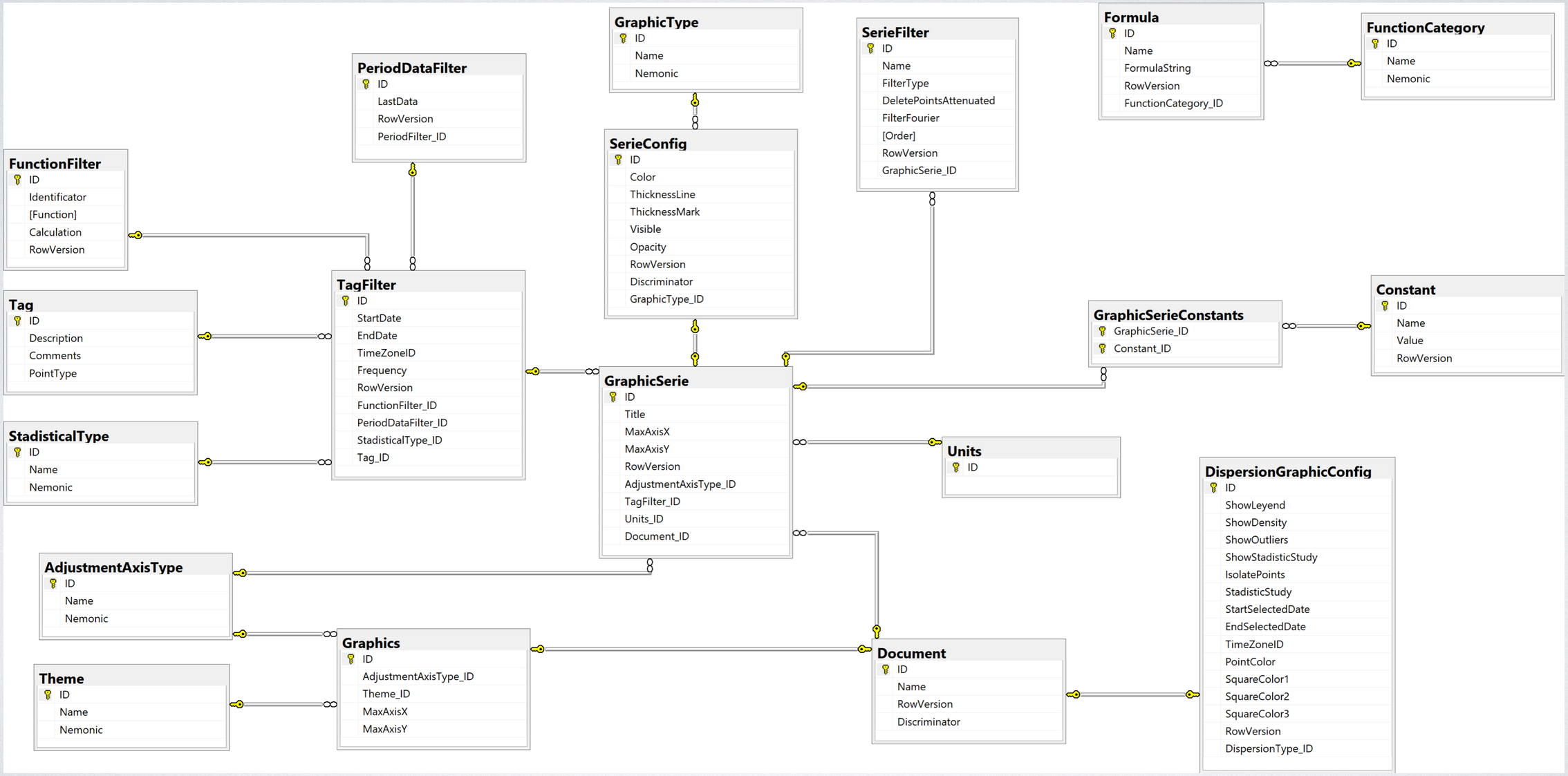


Diagrama SQL Simplificado

DISEÑO DETALLADO CONTROLADORES

Recurso	URI Relativo	Métodos
Documents	/api/Document	GET
		POST
Document	/api/Document/{ID}	GET
		PUT
		DELETE
AdjustmentAxisTypes	/api/AdjustmentAxisType	GET
AdjustmentAxisType	/api/AdjustmentAxisType/ByID/{ID}	GET
	/api/AdjustmentAxisType/ByNemonic/{Nemonic}	GET
GraphicTypes	/api/GraphicType	GET
GraphicType	/api/GraphicType/ByID/{ID}	GET
	/api/GraphicType/ByNemonic/{Nemonic}	GET
Themes	/api/Theme	GET
Theme	/api/Theme/ByID/{ID}	GET
	/api/Theme/ByNemonic/{Nemonic}	GET
CorrelationTypes	/api/CorrelationType	GET
CorrelationType	/api/CorrelationType/ByID/{ID}	GET
	/api/CorrelationType/ByNemonic/{Nemonic}	GET
DispersionTypes	/api/DispersionType	GET
DispersionType	/api/DispersionType/ByID/{ID}	GET
	/api/DispersionType/ByNemonic/{Nemonic}	GET
HeatColorMapTypes	/api/HeatColorMapType	GET
HeatColorMapType	/api/HeatColorMapType/ByID/{ID}	GET
	/api/HeatColorMapType/ByNemonic/{Nemonic}	GET
Formulas	/api/Formula	GET
		POST
Formula	/api/Formula/{ID}	POST
		PUT
		DELETE

FunctionCategories	/api/FunctionCategory	GET
FunctionCategory	/api/FunctionCategory/ByID/{ID}	GET
	/api/FunctionCategory/ByNemonic/{Nemonic}	GET
GroupedTypes	/api/GroupedType	GET
GroupedType	/api/GroupedType/ByID/{ID}	GET
	/api/GroupedType/ByNemonic/{Nemonic}	GET
SignalTypes	/api/SignalType	GET
SignalType	/api/SignalType/ByID/{ID}	GET
	/api/SignalType/ByNemonic/{Nemonic}	GET
StadisticalTypes	/api/StadisticalType	GET
StadisticalType	/api/StadisticalType/ByID/{ID}	GET
	/api/StadisticalType/ByNemonic/{Nemonic}	GET
Tags	/api/Tag	GET
Tag	/api/Tag/ByID/{ID}	GET
	/api/Tag/ByNemonic/{Nemonic}	GET
Units	/api/Units	GET
Unit	/api/Units/ByID/{ID}	GET
	/api/Units/ByNemonic/{Nemonic}	GET
PeriodUnits	/api/PeriodUnits	GET
PeriodUnit	/api/PeriodUnits/ByID/{ID}	GET
	/api/PeriodUnits/ByNemonic/{Nemonic}	GET
PredictionTypes	/api/PredictionType	GET
PredictionType	/api/PredictionType/ByID/{ID}	GET
	/api/PredictionType/ByNemonic/{Nemonic}	GET
TrendingTypes	/api/TrendingType	GET
TrendingType	/api/TrendingType/ByID/{ID}	GET
	/api/TrendingType/ByNemonic/{Nemonic}	GET

Definición de recursos, identificadores y métodos soportados

DISEÑO DETALLADO CONTROLADORES

Recurso	Método	Parámetros
Documents	GET	Name, DocumentType, GroupedTypeX, GroupedTypeY, PredictionType, TrendingType, CorrelationType, DispersionType, Tag, StadisticalType, GraphicType, LastID, PageSize
Formulas	GET	FunctionCategory
StadisticalTypes	GET	SignalType
Tags	GET	PointType

Definición de los parámetros soportados por cada método

Método	Código	Motivo
GET	200 OK	Recurso obtenido con éxito.
	404 NOT FOUND	No se encuentra el recurso.
POST	201 CREATED	Recurso creado con éxito.
	400 BAD REQUEST	La información contenida en el Body no es válida.
PUT	200 OK	Recurso modificado con éxito.
	404 NOT FOUND	No se encuentra el recurso.
	409 CONFLICT	Datos fuera de sincronización. El recurso ha sido modificado desde que fue leído.
DELETE	200 OK	Recurso eliminado con éxito.
	404 NOT FOUND	No se encuentra el recurso.

Definición de los códigos de respuesta

DISEÑO DETALLADO CONTROLADORES

package Design [ControllerLayer]

HeatColorMapTypeController

attributes
-heatColorMapTypeBusiness : IHeatColorMapTypeBusiness

operations
+HeatColorMapTypeController(heatColorMapTypeBusiness : IHeatColorMapTypeBusiness)
+GetByID(ID : long) : HttpResponseMessage
+GetByNemonic() : HttpResponseMessage
+List() : HttpResponseMessage

DispersionTypeController

attributes
-dispersionTypeBusiness : IDispersionTypeBusiness

operations
+DispersionTypeController(dispersionTypeBusiness : IDispersionTypeBusiness)
+GetByID(ID : long) : HttpResponseMessage
+GetByNemonic(nemonic : String) : HttpResponseMessage
+ListAll() : HttpResponseMessage

FormulaController

attributes
-formulaBusiness : IFormulaBusiness

operations
+FormulaController(formulaBusiness : IFormulaBusiness)
+Get(ID : long) : HttpResponseMessage
+Post(formula : Formula) : HttpResponseMessage
+Put(ID : long, formula : Formula) : HttpResponseMessage
+Delete(ID : long) : HttpResponseMessage
+List(functionCategory : String [0..1], orderBy : String [0..1], order : String [0..1]) : HttpResponseMessage

TrendingTypeController

attributes
-trendingTypeBusiness : ITrendingTypeBusiness

operations
+TrendingTypeController(trendingTypeBusiness : ITrendingTypeBusiness) : HttpResponseMessage
+GetByID(ID : long) : HttpResponseMessage
+GetByNemonic(nemonic : String) : HttpResponseMessage
+List() : HttpResponseMessage

UnitsController

attributes
-unitsBusiness : IUnitsBusiness

operations
+Units(unitsBusiness : IUnitsBusiness)
+Get(ID : long) : HttpResponseMessage
+List() : HttpResponseMessage

AdjustmentAxisTypeController

attributes
-adjustmentAxisTypeBusiness : IAdjustmentAxisTypeBusiness

operations
+AdjustmentAxisTypeController(adjustmentAxisTypeBusiness : IAdjustmentAxisTypeBusiness)
+GetByID(ID : long) : HttpResponseMessage
+GetByNemonic(nemonic : String) : HttpResponseMessage
+List() : HttpResponseMessage

StadisticalTypeController

attributes
-stadisticalTypeBusiness : IStadisticalTypeBusiness

operations
+StadisticalTypeController(stadisticalTypeBusiness : IStadisticalTypeBusiness)
+GetByID(ID : long) : HttpResponseMessage
+GetByNemonic(nemonic : String) : HttpResponseMessage
+List(signalType : String [0..1], orderBy : String [0..1], order : String [0..1]) : HttpResponseMessage

PeriodUnitController

attributes
-periodUnitBusiness : IPeriodUnitBusiness

operations
+PeriodUnitController(periodUnitBusiness : IPeriodUnitBusiness)
+GetByID(ID : long) : HttpResponseMessage
+GetByNemonic(nemonic : String) : HttpResponseMessage
+List() : HttpResponseMessage

PredictionTypeController

attributes
-predictionTypeBusiness : IPredictionTypeBusiness

operations
+PredictionTypeController(predictionTypeBusiness : IPredictionTypeBusiness)
+GetByID(ID : long) : HttpResponseMessage
+GetByNemonic(nemonic : String) : HttpResponseMessage
+List() : HttpResponseMessage

TagController

attributes
-tagBusiness : ITagBusiness

operations
+TagController(tagBusiness : ITagBusiness)
+GetByID() : HttpResponseMessage
+GetByNemonic(nemonic : String) : HttpResponseMessage
+List() : HttpResponseMessage

CorrelationTypeController

attributes
-correlationTypeBusiness : ICorrelationTypeBusiness

operations
+CorrelationTypeController(correlationTypeBusiness : ICorrelationTypeBusiness)
+GetByID(ID : long, nemonic : String) : HttpResponseMessage
+GetByNemonic(nemonic : String) : HttpResponseMessage
+List() : HttpResponseMessage

DocumentController

attributes
-documentBusiness : IDocumentBusiness

operations
+DocumentController(documentBusiness : IDocumentBusiness)
+Get(ID : long) : HttpResponseMessage
+Post(document : Document) : HttpResponseMessage
+Put(ID : long, document : Document) : HttpResponseMessage
+Delete(ID : long) : HttpResponseMessage
+List(filters : FilterDTO [0..1]) : HttpResponseMessage

SignalTypeController

attributes
-signalTypeBusiness : ISignalTypeBusiness

operations
+SignalTypeController(signalTypeBusiness : ISignalTypeBusiness)
+GetByID(ID : long) : HttpResponseMessage
+GetByNemonic(nemonic : String) : HttpResponseMessage
+List() : HttpResponseMessage

FunctionCategoryController

attributes
-functionCategoryBusiness : IFunctionCategoryBusiness

operations
+FunctionCategoryController(functionCategoryBusiness : IFunctionCategoryBusiness)
+GetByID(ID : long) : HttpResponseMessage
+GetByNemonic(nemonic : String) : HttpResponseMessage
+List() : HttpResponseMessage

GroupedTypeController

attributes
-groupedTypeBusiness : IGroupedTypeBusiness

operations
+GroupedTypeController(groupedTypeBusiness : IGroupedTypeBusiness)
+GetByID(ID : long) : HttpResponseMessage
+GetByNemonic(nemonic : String) : HttpResponseMessage
+List() : HttpResponseMessage

ThemeController

attributes
-themeBusiness : IThemeBusiness

operations
+ThemeController(themeBusiness : IThemeBusiness)
+GetById(ID : long) : HttpResponseMessage
+GetByNemonic(nemonic : String) : HttpResponseMessage
+List() : HttpResponseMessage

GraphicTypeController

attributes
-graphicTypeBusiness : IGraphicTypeBusiness

operations
+GraphicTypeController(graphicTypeBusiness : IGraphicTypeBusiness)
+GetByID(ID : long) : HttpResponseMessage
+GetByNemonic(nemonic : String) : HttpResponseMessage
+List() : HttpResponseMessage

ÍNDICE

Contexto
Motivación TFG
Objetivos del Proyecto
Metodología
Presentación del Problema
Requisitos
Diseño Arquitectónico
Diseño Detallado
Pruebas
Demo
Conclusiones
Trabajos Futuros

PRUEBAS

- Pruebas Unitarias
- Pruebas de Integración

Cobertura de sentencias, condiciones y decisiones

- Pruebas de Sistema
  - ***Pruebas de Rendimiento***: Tiempos de Respuesta + Uso de Recursos
  - ***Pruebas de Mantenibilidad***: MySQL
  - ***Pruebas de Seguridad:*** Inyección SQL
- Pruebas de Aceptación

ÍNDICE

Contexto
Motivación TFG
Objetivos del Proyecto
Metodología
Presentación del Problema
Requisitos
Diseño Arquitectónico
Diseño Detallado
Pruebas
Demo
Conclusiones
Trabajos Futuros

DEMO

ÍNDICE

Contexto
Motivación TFG
Objetivos del Proyecto
Metodología
Presentación del Problema
Requisitos
Diseño Arquitectónico
Diseño Detallado
Pruebas
Demo
Conclusiones
Trabajos Futuros

CONCLUSIONES

- Definición del modelo de Base de Datos (BBDD).
- Diseño e implementación del acceso a BBDD para la recuperación y persistencia de las gráficas.
- Diseño e implementación de todo el negocio relacionado con las gráficas de IDbox.
- Diseño e implementación de los servicios que ofrecerán las distintas funcionalidades gráficas de IDbox.
- Diseño e implementación de una Web API para el consumo externo de gráficas.

ÍNDICE

Contexto
Motivación TFG
Objetivos del Proyecto
Metodología
Presentación del Problema
Requisitos
Diseño Arquitectónico
Diseño Detallado
Pruebas
Demo
Conclusiones
Trabajos Futuros

TRABAJOS FUTUROS

- Optimización
  - Consultas SQL
  - Paralelización de consultas
- Pruebas de Rendimiento
  - Entorno de pruebas más cercano al entorno de despliegue
  - Servidor BBDD + Servidores Web API + Balanceador de Carga
- Seguridad
- Incorporación de nuevas características

DEFINICIÓN Y DESARROLLO DEL MÓDULO GRÁFICO DE IDbox

(REDESIGNING IDbox's GRAPHIC MODULE)

Isaac Gutiérrez Rodríguez

Email: igr895@gmail.com