



**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)**  
**MÁSTER EN BIG DATA: TECNOLOGÍA Y ANALÍTICA AVANZADA**

**DISEÑO DE UNA ARQUITECTURA DE  
PRESUPUESTACIÓN Y REPORTING CORPORATIVO  
BASADA EN ORACLE CLOUD EPM PLANNING**

Trabajo de Fin de Máster

**Autor:** Alejandro Coman Venceslá

**Director:** Luis Álvaro González García-Calvo

Madrid

Junio 2026

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

**DISEÑO DE UNA ARQUITECTURA DE  
PRESUPUESTACIÓN Y REPORTING CORPORATIVO  
BASADA EN ORACLE CLOUD EPM PLANNING**

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el  
curso académico 2025/26 es de mi autoría, original e inédito y  
no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido  
tomada de otros documentos está debidamente referenciada.

Fdo.: Alejandro Coman Venceslá      Fecha: 05/06/2026

Autorizada la entrega del proyecto EL DIRECTOR DEL PROYECTO

  
*Luis Álvaro González García-Calvo*

Fdo.: Luis Álvaro González García-Calvo      Fecha: Fecha: 05/06/2026

Vº Bº del Coordinador de Proyectos

Fdo.: Carlos Morrás Ruiz-Falcó      Fecha: 05/06/2026

## **AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO**

### **1º. Declaración de la autoría y acreditación de la misma.**

El autor D. **Alejandro Coman Venceslá** DECLARA ser el titular de los derechos de propiedad intelectual de la obra: **Diseño de una Arquitectura de Presupuestación y Reporting Corporativo Basada en Oracle Cloud EPM Planning**, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

### **2º. Objeto y fines de la cesión.**

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

### **3º. Condiciones de la cesión y acceso**

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducir la en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

### **4º. Derechos del autor.**

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

### **5º. Deberes del autor.**

- El autor se compromete a:
  - a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
  - b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
  - c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
  - d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción

de derechos derivada de las obras objeto de la cesión.

**6º. Fines y funcionamiento del Repositorio Institucional.**

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 05. De junio de 2026

**ACEPTA**

Fdo.....

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

# Resumen

---

Los modelos financieros basados en hojas de cálculo y procesos manuales generan silos de información, versiones contradictorias del mismo dato y cierres lentos que restan tiempo al análisis. Este trabajo define una **arquitectura funcional** de presupuestación y reporting corporativo —sobre **Oracle Cloud EPM Planning** y **Narrative Reporting**, tras comparar las soluciones líderes del mercado— orientada a tres problemas: ganar **eficiencia operativa** en la carga y el cierre, asegurar la **integridad** de las cifras que circulan en el grupo y dotar a finanzas de **agilidad** para actualizar previsiones y escenarios con menor latencia.

La arquitectura combina un modelo de datos común por filial y escenario, integración automática desde el ERP, un protocolo de versiones que separa carga, ajustes y dato validado, variables de sustitución que unifican año, mes y escenario activo en cada ciclo, listas de tareas que guían al planificador, reglas de cálculo parametrizadas y controles de acceso por entidad. Esas piezas permiten sustituir extracciones repetitivas, evitar que el «número oficial» quede en duda, cerrar el *Real* y el *Forecast* con trazabilidad y publicar informes ejecutivos alineados con el dato aprobado, sin reconstruir presentaciones desde cero. En conjunto, la propuesta sitúa a la función financiera como *business partner*: menos reconciliación manual y más capacidad para interpretar desviaciones y decidir.

# Palabras clave

---

Las palabras clave recogen el alcance del trabajo desde la función financiera, los procesos que la arquitectura cubre y la plataforma en la que se apoya.

<b>Función financiera y procesos</b>	Planificación financiera; FP&A; presupuestación corporativa; <i>rolling forecast</i> ; escenarios <i>what-if</i> ; cierre contable; conciliación; <i>business partner</i> financiero; transformación de la función financiera.
<b>Problemas y objetivos</b>	Eficiencia operativa; integridad del dato; agilidad estratégica; automatización de procesos; fuente única de verdad; eliminación de silos; reporting ejecutivo; decisiones basadas en datos.
<b>Plataforma y componentes</b>	Oracle Cloud EPM Planning; Narrative Reporting; Essbase; integración ERP; Data Integration; variables de sustitución; <i>runtime prompts</i> ; listas de tareas; lógica de negocio; IPM Insights.
<b>Gobierno y seguridad</b>	Gobierno de metadatos; alias; ciclo de vida del dato; control de versiones; validación entre versiones; seguridad por entidad.

# Índice de figuras

2.1	Cuadrante Mágico de Gartner para el Software de Planificación Financiera (diciembre 2025). Fuente: Crowder y Sahoo 2025. . . . .	6
3.1	Arquitectura global propuesta para la presupuestación y el reporting corporativo sobre Oracle Cloud EPM. . . . .	14
6.1	Ciclo de vida del dato financiero desde la carga inicial hasta la validación y consumo en reporting. . . . .	37
7.1	Flujos intraescenario e interescenario entre versiones de trabajo, validación y planificación. . . . .	41
8.1	Orquestación de contexto entre formulario, Groovy, RTPs y Calc Scripts. . . . .	47

# Índice de tablas

2.1	Tabla comparativa de las principales plataformas de planificación financiera bajo los ejes Gartner 2025. Elaboración propia a partir de Crowder y Sahoo 2025. . . . .	10
4.1	Comparativa de tipos de almacenamiento Essbase para planificación.	20
4.2	Operadores de consolidación Essbase y uso en la arquitectura (Oracle Corporation 2025h). . . . .	26
4.3	Política de almacenamiento por tipo de miembro. . . . .	26
4.4	Plantilla genérica de jerarquía para dimensiones personalizadas. . .	28
4.5	Raíz, _DF, _NA y patrón de hojas por dimensión. . . . .	29
6.1	Fuentes de datos y tipos de conexión en la arquitectura de integración.	35
6.2	Tipos de mapeo de miembros en Data Integration (Oracle Corporation 2026j). . . . .	36
6.3	Propiedades del protocolo de versiones V00 / V01 / V02 (Oracle Corporation 2026j). . . . .	37
7.1	Familias de formularios en la arquitectura operativa. . . . .	39
7.2	Variables de sustitución mínimas para parametrizar los flujos operativos. . . . .	41
7.3	Tipos de tarea y su papel en las listas de tareas de la arquitectura. .	42
7.4	Flujos de ejecución diferenciados por tipo de dato. . . . .	42
8.1	Alcance dimensional habitual de las reglas Calc (filtro FIX). . . . .	45
8.2	Cuándo usar Calc Script, Groovy o Rule Set en la arquitectura. . . .	48
9.1	Papel de IPM Insights, Auto Predict y BYOML en la arquitectura predictiva. . . . .	50
10.1	Roles estándar de Oracle y su encaje en la arquitectura propuesta. .	55
10.2	Matriz de permisos por rol y artefacto en la arquitectura propuesta.	56
10.3	Política de seguridad por versión dentro del ciclo de vida del dato.	58
11.1	Plantilla de secciones y <i>Doclets</i> de un Report Package de informe de resultados (periodicidad configurable). . . . .	62

# Índice de listados de código

---

A.1	Plantilla genérica de cabecera (comentarios al inicio de cada regla)	74
A.2	Copiado de versión: CLEARBLOCK en destino y DATACOPY V00 → V01	75
A.3	Cálculo con fórmula: CLEARBLOCK del destino y IF/ENDIF en V00 .	76
A.4	Validación en guardado ( <i>Run After Save</i> ) . . . . .	77
A.5	Orquestación desde menú de acción: lanzar Calc Script vía API REST	78
A.6	Orquestación desde rejilla: POST REST por celda KPI válida . . . . .	79
A.7	Smart Push acotado por entidad y período del formulario activo . .	80
A.8	Envío filial → grupo: una porción con createSmartPush . . . . .	81

# Índice general

<b>Resumen</b>	<b>I</b>
<b>Palabras clave</b>	<b>II</b>
<b>Índice de figuras</b>	<b>III</b>
<b>Índice de tablas</b>	<b>v</b>
<b>Índice de listados de código</b>	<b>v</b>
<b>1 Introducción y Motivación</b>	<b>1</b>
1.1 Introducción del proyecto . . . . .	1
1.2 Motivación . . . . .	2
1.2.1 Eficiencia operativa . . . . .	2
1.2.2 Integridad del dato . . . . .	2
1.2.3 Agilidad estratégica . . . . .	3
<b>2 Estado del Arte y Análisis Comparativo</b>	<b>4</b>
2.1 Panorama del software de planificación financiera . . . . .	4
2.2 El Cuadrante Mágico de Gartner: diciembre de 2025 . . . . .	5
2.3 Ejes de análisis: tendencias del mercado como criterio comparativo	6
2.4 Análisis comparativo de las soluciones líderes . . . . .	7
2.4.1 OneStream . . . . .	7
2.4.2 Anaplan . . . . .	7
2.4.3 SAP Analytics Cloud . . . . .	8
2.4.4 Oracle Cloud EPM Planning . . . . .	8
2.4.5 Tabla comparativa resumida . . . . .	9
2.5 Justificación de Oracle Cloud EPM como plataforma base . . . . .	9
<b>3 Metodología de Investigación Basada en el Diseño</b>	<b>12</b>
3.1 Enfoque metodológico . . . . .	12
3.1.1 Criterio de documentación . . . . .	13

3.1.2	Uso asistido de inteligencia artificial generativa . . . . .	13
3.2	Secuencia de construcción del artefacto . . . . .	14
3.2.1	El modelo multidimensional como base del artefacto . . . . .	15
3.2.2	El ciclo de vida del dato como capa de control . . . . .	15
3.2.3	La orquestación avanzada como capa de consumo y ejecución . . . . .	16
3.3	Demostración y evaluación del diseño . . . . .	17
<b>4</b>	<b>Diseño del Modelo de Datos Multidimensional</b>	<b>19</b>
4.1	Selección del tipo de cubo: BSO, ASO e Híbrido . . . . .	19
4.1.1	Block Storage Option (BSO) . . . . .	19
4.1.2	Aggregate Storage Option (ASO) . . . . .	19
4.1.3	Modo Híbrido: decisión adoptada . . . . .	20
4.2	Diseño de las dimensiones . . . . .	20
4.2.1	Dimensión Mes (Period) . . . . .	20
4.2.2	Dimensión Escenario . . . . .	21
4.2.3	Dimensión Versión . . . . .	22
4.2.4	Dimensión Year (Año) . . . . .	22
4.2.5	Dimensión Entidad . . . . .	23
4.2.6	Dimensión Estados Financieros (Account) . . . . .	23
4.2.7	Dimensión Divisa (Currency) — opcional . . . . .	24
4.2.8	Dimensiones personalizadas . . . . .	24
4.3	Configuración técnica del modelo . . . . .	25
4.3.1	Estructura de jerarquías: generaciones, niveles y miembros compartidos . . . . .	25
4.3.2	Operadores de agregación . . . . .	26
4.3.3	Propiedades de almacenamiento de miembros . . . . .	26
4.3.4	Clasificación Dense/Sparse de las dimensiones . . . . .	27
4.4	Política de nomenclatura y convenciones . . . . .	27
4.4.1	Restricciones técnicas de Oracle . . . . .	27
4.4.2	Convenciones generales de la arquitectura . . . . .	27
4.4.3	Alias de miembro: nombre técnico y etiqueta de presentación . . . . .	28
4.4.4	Plantilla genérica de jerarquía . . . . .	28
4.4.5	Convenciones específicas por dimensión . . . . .	28
<b>5</b>	<b>Gestión y Gobierno de Metadatos en Oracle Cloud EPM Planning</b>	<b>30</b>
5.1	Herramientas de mantenimiento en Planning . . . . .	30
5.1.1	Importación de metadatos por fichero . . . . .	30
5.1.2	Editor puntual y EPM Automate . . . . .	31
5.2	Gobierno del proceso de cambio . . . . .	31

5.2.1	Roles y responsabilidades . . . . .	31
5.2.2	Procedimiento de cambio estructural . . . . .	31
5.3	Congelación de metadatos durante el ciclo presupuestario . . . . .	32
<b>6</b>	<b>Arquitectura de Integración y Ciclo de Vida del Dato</b>	<b>34</b>
6.1	Componentes de la arquitectura de integración . . . . .	34
6.1.1	EPM Integration Agent . . . . .	34
6.1.2	Módulo Data Integration . . . . .	34
6.1.3	Fuentes de datos . . . . .	34
6.2	Configuración de una regla de integración . . . . .	34
6.2.1	Localizaciones ( <i>Locations</i> ) . . . . .	34
6.2.2	Formato de importación y consulta SQL . . . . .	35
6.2.3	Filtros de ejecución . . . . .	35
6.2.4	Mapeos de miembros ( <i>Member Mappings</i> ) . . . . .	35
6.2.5	Transformaciones . . . . .	36
6.3	Ejecución y validación . . . . .	36
6.4	Ciclo de vida del dato: versiones V00, V01 y V02 . . . . .	36
6.5	Matrices de validación entre versiones . . . . .	37
6.6	Gestión de errores y controles de calidad . . . . .	38
6.6.1	Controles preventivos . . . . .	38
6.6.2	Controles detectivos . . . . .	38
<b>7</b>	<b>Proceso Operativo de Planificación</b>	<b>39</b>
7.1	Formularios . . . . .	39
7.1.1	Tipos de formulario en la arquitectura . . . . .	39
7.1.2	Lógica asociada al formulario . . . . .	39
7.2	Listas de tareas y orquestación del proceso . . . . .	40
7.2.1	Listas de tareas ( <i>Task Lists</i> ) . . . . .	40
7.2.2	Tipos de tarea disponibles . . . . .	40
7.3	Flujos de ejecución por tipo de dato . . . . .	40
7.3.1	Variables de sustitución y parametrización de flujos . . . . .	41
7.3.2	Composición de las listas de tareas . . . . .	42
7.3.3	Flujo del Real: carga y conciliación . . . . .	42
7.3.4	Flujo del Forecast: avance mensual . . . . .	42
7.3.5	Flujo de la Proyección a largo plazo . . . . .	43
7.3.6	Instanciación por entidad: holding y filiales . . . . .	43
<b>8</b>	<b>Lógica de Negocio Avanzada y Orquestación</b>	<b>44</b>
8.1	Calc Scripts: lógica de cálculo masivo . . . . .	44

8.1.1	Cómo estructurar los Calc Scripts . . . . .	44
8.1.2	Cabecera, dependencias y <i>changelog</i> . . . . .	45
8.1.3	Qué trozo del cubo toca cada regla (el FIX) . . . . .	45
8.1.4	Promoción entre versiones (V00 a V01) . . . . .	45
8.2	Paso de contexto entre formulario, Groovy y Calc Script . . . . .	46
8.3	Groovy Scripts: orquestación dinámica en tiempo real . . . . .	47
8.3.1	Validación al guardar el formulario . . . . .	47
8.3.2	Lanzar reglas Calc desde el formulario . . . . .	47
8.3.3	Una misma regla para varios años y escenarios . . . . .	48
8.3.4	Envío de datos entre cubos o entre filiales . . . . .	48
8.4	Cuándo usar Calc Script, Groovy o Rule Set . . . . .	48
<b>9</b>	<b>Analítica Predictiva e Inteligencia Artificial con IPM Insights</b>	<b>49</b>
9.1	Arquitectura de IPM Insights . . . . .	49
9.1.1	Configuración inicial del motor . . . . .	49
9.1.2	Justificación metodológica de los parámetros predictivos . . . . .	50
9.2	Caso de uso 1: Detección de sesgos en el proceso presupuestario . . . . .	51
9.2.1	Descripción del problema . . . . .	51
9.2.2	Implementación con IPM Insights . . . . .	52
9.2.3	Uso operativo del análisis de sesgo . . . . .	52
9.3	Caso de uso 2: Previsión automática de ingresos por series temporales . . . . .	52
9.3.1	Descripción del problema . . . . .	52
9.3.2	Implementación con Auto Predict e IPM Insights . . . . .	52
9.3.3	Escritura de predicciones en el cubo . . . . .	53
9.4	Caso de uso 3: Detección de anomalías en la carga de datos . . . . .	53
9.4.1	Descripción del problema . . . . .	53
9.4.2	Implementación . . . . .	53
9.5	Integración de modelos externos mediante BYOML . . . . .	54
9.6	Limitaciones y validación humana . . . . .	54
<b>10</b>	<b>Modelo de Seguridad y Gobierno de Accesos</b>	<b>55</b>
10.1	Roles estándar de Oracle Cloud EPM . . . . .	55
10.2	Matriz de roles y permisos por artefacto . . . . .	56
10.3	Grupos funcionales corporativos . . . . .	56
10.3.1	Diseño de grupos propuestos . . . . .	56
10.3.2	Principios de gestión de grupos . . . . .	57
10.4	Seguridad de datos por versión, escenario y entidad . . . . .	57
10.5	Intersecciones Válidas . . . . .	58
10.5.1	Funcionamiento técnico . . . . .	58

10.5.2 Ejemplos de reglas propuestas . . . . .	58
10.6 Segregación de funciones y auditoría . . . . .	59
<b>11 Reporting Ejecutivo y Narrative Reporting</b>	<b>60</b>
11.1 Reporting operacional en Oracle Cloud EPM Planning . . . . .	60
11.1.1 Informes ad-hoc y dashboards . . . . .	60
11.2 Arquitectura de Oracle Narrative Reporting . . . . .	61
11.2.1 Conexión federada con Planning . . . . .	61
11.3 Report Package y flujo de elaboración . . . . .	61
11.4 Ventaja frente al reporting manual . . . . .	61
<b>12 Conclusiones y Líneas Futuras</b>	<b>63</b>
12.1 Cumplimiento de los objetivos . . . . .	63
12.1.1 Eficiencia operativa . . . . .	63
12.1.2 Integridad del dato . . . . .	63
12.1.3 Agilidad estratégica . . . . .	64
12.2 Valoración crítica de la arquitectura . . . . .	64
12.2.1 Fortalezas . . . . .	64
12.2.2 Limitaciones y riesgos . . . . .	64
12.3 Líneas de trabajo futuro . . . . .	65
12.3.1 Orquestación corporativa con EPM Task Manager . . . . .	65
12.3.2 Gobierno de metadatos (EDMCS) . . . . .	65
12.3.3 Extensión a la consolidación legal (Oracle Financial Consolidation and Close) . . . . .	65
12.3.4 Integración con arquitectura Data Mesh . . . . .	65
12.3.5 Modelos de lenguaje generativo (LLM) sobre los cubos de planificación . . . . .	65
12.4 Reflexión final . . . . .	66
<b>Bibliografía y fuentes consultadas</b>	<b>67</b>
<b>A Ejemplos de código</b>	<b>73</b>
A.1 Plantilla de cabecera (Calc Script y Groovy) . . . . .	73
A.2 Calc Scripts . . . . .	75
A.3 Groovy Scripts . . . . .	77

## Capítulo 1

# Introducción y Motivación

---

## 1.1 Introducción del proyecto

---

Hablar de transformación digital en las finanzas corporativas ya no es hablar del futuro: es describir algo que está ocurriendo, con distinto grado de madurez, en prácticamente cualquier organización de cierto tamaño. Lo que sí sigue siendo un problema vigente es la brecha entre la velocidad a la que se generan los datos y la velocidad a la que los equipos financieros son capaces de procesarlos y convertirlos en decisiones.

Los datos crecen. Los plazos para tomar decisiones, sin embargo, se acortan. Esa tensión es el punto de partida de este trabajo.

Las arquitecturas financieras tradicionales, aún ampliamente usadas, se caracterizan por la presencia de silos de información y por una dependencia excesiva de hojas de cálculo y procesos manuales. Esta fragmentación compromete la calidad de la información y limita la capacidad del departamento financiero para actuar como socio estratégico del negocio (*business partner*). Según FTI Consulting (FTI Consulting 2025), el 82 % de los directores financieros identifica precisamente la complejidad de los procesos manuales como el principal freno de su función, y el 79 % declara tener previsto invertir en actualizaciones de software FP&A a lo largo de 2025.

Ante este escenario, las organizaciones buscan alternativas que sustituyan o complementen los modelos convencionales. Las plataformas modernas de planificación financiera en la nube (*Financial Planning & Analysis, FP&A*) ofrecen un modelo de datos común, colaboración en tiempo real y automatización de flujos de trabajo de principio a fin, permitiendo integrar datos desde diversas fuentes —sistemas ERP (*Enterprise Resource Planning*), bases de datos relacionales, soluciones de BI (*Business Intelligence*) o archivos locales— crear informes avanzados automatizados y establecer sistemas de gobierno del dato más rigurosos.

La transición a estos modelos acerca con menor latencia las decisiones de áreas operativas (ventas, RR. HH.) a las proyecciones financieras globales y al impacto en los estados financieros. Este trabajo diseña una arquitectura de presupuestación y reporting corporativo; la plataforma tecnológica se justifica en el capítulo 2, tras el

análisis de mercado y la comparativa de soluciones líderes.

## 1.2 Motivación

Partiendo del contexto anterior, la motivación de este proyecto está fundamentada en la necesidad de solventar las deficiencias críticas de los modelos financieros tradicionales. Aunque estos modelos están ampliamente extendidos en el tejido empresarial, están siendo complementados o reemplazados progresivamente por soluciones más integradas, automatizadas y orientadas al dato. La transformación que se persigue gira en torno a tres ejes fundamentales: la eficiencia operativa, la integridad del dato y la agilidad estratégica.

### 1.2.1 Eficiencia operativa

Gran parte del tiempo de los equipos financieros se dedica actualmente a la recolección, limpieza y consolidación manual de datos, restando capacidad al análisis y a la toma de decisiones, donde reside realmente el valor estratégico. Una encuesta global de McKinsey (McKinsey & Company 2024) revela que, aunque el 98 % de los departamentos financieros ha invertido en digitalización y automatización, únicamente el 25 % o menos de sus procesos está actualmente automatizado: la inversión existe, pero su alcance real sigue siendo limitado. La tendencia se mantiene en 2026: el informe de FTI Consulting de ese año (FTI Consulting 2026) señala que la automatización de la previsión financiera y del reporting son las dos capacidades de IA que los directores financieros priorizan de forma más inmediata, con un 90 % y un 88 % de menciones respectivamente. Uno de los objetivos centrales de este proyecto es precisamente reducir esa brecha, automatizando las tareas operativas de mayor latencia para que el equipo financiero pueda desplazarse desde el rol de «procesador de datos» al de «analista estratégico».

### 1.2.2 Integridad del dato

Los silos de información generan incongruencias entre departamentos, erosionan la confianza en los reportes corporativos y plantean la pregunta recurrente de qué dato es el correcto. Una plataforma de planificación con un modelo de datos común y reglas de gobierno sobre jerarquías y versiones permite mantener una estructura unificada y auditable. Esto garantiza que la información financiera sea transparente y coherente en todos los niveles de la organización, reduciendo el riesgo de versiones paralelas del «mismo» número circulando sin control.

### 1.2.3 Agilidad estratégica

En un entorno global y volátil, los presupuestos anuales estáticos han perdido relevancia operativa. Las empresas necesitan generar escenarios dinámicos (*What-if*) y previsiones continuas (*Rolling Forecasts*), así como integrar analítica predictiva basada en inteligencia artificial para anticiparse a las tendencias de mercado. Esta necesidad ya no es percibida como una aspiración futura: el 87 % de los directores financieros encuestados por Deloitte espera que la IA sea extremadamente importante para las operaciones de su departamento en 2026, y el 50 % cita la transformación digital de la función financiera como su prioridad estratégica número uno para ese año (Deloitte 2025).

La evidencia de 2026 comienza a cuantificar el impacto de esta apuesta. Un informe de KPMG (KPMG 2026), elaborado a partir de 1.013 directivos financieros en 13 sectores y 20 países, constata que las organizaciones que han desplegado IA agéntica en sus flujos de planificación financiera reportan una mejora de al menos 32 puntos porcentuales en sus métricas de rendimiento clave, con un incremento de casi 40 puntos en la precisión de sus previsiones. La conclusión del informe es significativa: la IA en finanzas está operando como motor de decisión, no como palanca de reducción de costes. El capítulo 2 recoge cómo estas exigencias se traducen en capacidades concretas del software FP&A avanzado y qué criterios permiten seleccionar una plataforma acorde al alcance de este trabajo.

## Capítulo 2

# Estado del Arte y Análisis Comparativo

---

La transformación digital de *Financial Planning & Analysis* (FP&A) ya forma parte del mercado operativo. El segmento de plataformas de gestión del rendimiento empresarial (*Enterprise Performance Management* – EPM) alcanzará los 7.690 millones de dólares en 2026 y se proyecta hasta los 11.890 millones en 2031, con una tasa compuesta anual del 9,1 % (Mordor Intelligence 2026). El crecimiento procede, sobre todo, de la adopción cloud y de la integración de inteligencia artificial en los procesos de planificación.

Este capítulo recorre ese contexto en tres pasos: primero, el estado actual del software de planificación financiera; después, el marco comparativo que ofrece el Cuadrante Mágico de Gartner; y, por último, la decisión de plataforma para la arquitectura propuesta en este TFM, que se formaliza en el apartado 2.5.

## 2.1 Panorama del software de planificación financiera

---

Durante décadas, la planificación y el reporting corporativo se apoyaron en hojas de cálculo, bases departamentales y cierres manuales. Ese modelo ya no responde a entornos multientidad, multidivisa y multicanal, donde finanzas necesita repositorio único, trazabilidad de versiones, seguridad por rol y recálculo de escenarios sin reconstruir modelos.

El software FP&A avanzado cubre precisamente ese vacío. Sus plataformas comparten varias capacidades que han pasado de ser diferenciadores a ser requisitos de mercado:

- **Modelo multidimensional:** Los datos se organizan en dimensiones (período, escenario, entidad, cuenta, etc.) que permiten agregar y desagregar con coherencia, en lugar de depender de tablas planas en Excel.
- **Integración con el ERP y otras fuentes:** Conectores, agentes de integración o APIs sustituyen las extracciones manuales y reducen el riesgo de error en la carga de datos reales y maestros.

- **Workflow y colaboración:** Formularios web, listas de tareas y flujos de aprobación coordinan la entrada de datos entre finanzas y áreas operativas.
- **Lógica de negocio automatizada:** Reglas de cálculo, validaciones y propagaciones entre escenarios liberan tiempo del equipo frente a consolidaciones repetitivas en hojas de cálculo.
- **Reporting y analítica:** Informes operacionales, cuadros de mando y, en los despliegues más maduros, reporting narrativo cualitativo para la alta dirección.
- **Inteligencia artificial embebida:** Detección de anomalías, previsión por series temporales y, de forma creciente, asistentes generativos o agentes que apoyan el análisis y la redacción de informes.

El Cuadrante Mágico de Gartner de diciembre de 2025 (Crowder y Sahoo 2025) confirma tres vectores de presión: IA generativa y agéntica, conectividad con plataformas cloud de datos y expansión hacia planificación integrada de negocio (*Integrated Business Planning – IBP*). Los apartados siguientes comparan cómo responden los principales proveedores.

## 2.2 El Cuadrante Mágico de Gartner: diciembre de 2025

El informe de diciembre de 2025 evalúa catorce plataformas de planificación financiera, clasificándolas en función de dos dimensiones: *Ability to Execute* (capacidad de ejecución actual) y *Completeness of Vision* (amplitud y solidez de la visión estratégica) (Crowder y Sahoo 2025). El cuadrante resultante sitúa a ocho vendedores en el cuadrante de Líderes: **Anaplan, Board, Jedox, OneStream, Oracle, SAP, Wolters Kluwer** y **Workday**. Dos plataformas se posicionan como Retadores (*Challengers*), dos como Visionarios (*Visionaries*) y dos en nichos especializados (*Niche Players*).

Para el análisis comparativo de este TFM se seleccionan los cuatro líderes que resultan más relevantes por su implantación en grandes corporaciones europeas y por la amplitud de su propuesta funcional: **Oracle Cloud EPM, OneStream, Anaplan** y **SAP Analytics Cloud**. La elección de este subconjunto responde a criterios de representatividad del mercado objetivo: las cuatro plataformas concentran la mayor parte de los proyectos de transformación FP&A en empresas con ingresos superiores a 500 millones de euros.



Gartner

**Figura 2.1:** Cuadrante Mágico de Gartner para el Software de Planificación Financiera (diciembre 2025). Fuente: Crowder y Sahoo 2025.

## 2.3 Ejes de análisis: tendencias del mercado como criterio comparativo

Los cuatro ejes de comparación se derivan de las tendencias que Gartner identifica para 2025–2026 (Crowder y Sahoo 2025):

**Eje 1 – Motor de cálculo y arquitectura:** Capacidad para gestionar modelos de gran escala (*large-scale modeling*), alta concurrencia y tiempos de respuesta estables.

**Eje 2 – IA embebida, analítica predictiva y GenAI:** Series temporales, anomalías, simulación estocástica, narrativas generativas, consultas en lenguaje natural, agentes autónomos, explicabilidad (*XAI*) y drivers externos.

**Eje 3 – Integración de datos y conectividad empresarial:** Conectores ERP/HCM/CRM,

mapeo *low-code* y acceso a arquitecturas cloud como Snowflake, Microsoft Fabric, Databricks o lakehouse.

**Eje 4 – IBP y colaboración interfuncional:** Convergencia entre planificación financiera y operativa: RRHH, supply chain, marketing y ESG, con propagación automática entre planes.

## 2.4 Análisis comparativo de las soluciones líderes

### 2.4.1 OneStream

**Motor de cálculo y arquitectura:** OneStream se sustenta en *Extensible Dimensionality*: un cubo corporativo común convive con cubos específicos de menor granularidad que heredan la estructura central. La consolidación opera en memoria con procesamiento selectivo de datos modificados (*dirty data*), evitando recálculos totales.

**IA embebida, analítica predictiva y GenAI:** El ecosistema *Sensible AI* integra algoritmos de Machine Learning *ML* directamente en el flujo de trabajo financiero, cubriendo previsión de series temporales, detección de sesgos humanos en proyecciones y explicabilidad (*XAI*). Gartner destaca que OneStream ha avanzado significativamente en capacidades de IA generativa para la narrativa financiera automatizada.

**Integración de datos y conectividad empresarial:** Combina conectores nativos para ERP (SAP, Oracle EBS, Workday), procesamiento *Map-and-Load* y consultas directas a Data Lakes para no saturar los cubos de consolidación.

**IBP y colaboración interfuncional:** Dispone de módulos para personal y proyectos integrados con el plan financiero. La propagación entre dimensiones es sólida, aunque supply chain y ESG quedan menos desarrollados que en otros líderes.

### 2.4.2 Anaplan

**Motor de cálculo y arquitectura:** Anaplan usa *Hyperblock*, motor en memoria basado en celdas vinculadas. Los cambios en *drivers* se propagan instantáneamente, lo que favorece escenarios *What-if*; su límite relativo está en consolidación financiera de alta complejidad.

**IA embebida, analítica predictiva y GenAI:** *Anaplan CoPlanner* permite interactuar con modelos en lenguaje natural. La predictiva se centra en demanda

y validación presupuestaria; los drivers externos requieren API y no existe biblioteca certificada *out-of-the-box*.

**Integración de datos y conectividad empresarial:** Es una plataforma abierta con conectores a ERPs y Data Lakes mediante *Anaplan Connect* y APIs REST. Gartner señala que ha mejorado su conectividad con plataformas cloud de datos, aunque la certificación de conectores a entornos específicos sigue siendo inferior a la de algunos competidores.

**IBP y colaboración interfuncional:** Es su principal fortaleza. La arquitectura *xP&A* conecta finanzas, RRHH, supply chain y marketing en tiempo real; Gartner la sitúa como referencia en planificación conectada.

### 2.4.3 SAP Analytics Cloud

**Motor de cálculo y arquitectura:** SAP Analytics Cloud opera sobre *SAP HANA Cloud*, motor columnar en memoria. Su «modelo único» unifica analítica y planificación, reduciendo movimiento de datos, pero maximiza rendimiento dentro del ecosistema SAP.

**IA embebida, analítica predictiva y GenAI:** Incorpora *Smart Predict* para la generación automatizada de previsiones estadísticas, y *Joule*, el copiloto de IA generativa de SAP, permite interactuar con los datos en lenguaje natural. La integración de IA está condicionada por la disponibilidad de datos en el ecosistema S/4HANA; las fuentes externas requieren pasos adicionales de integración.

**Integración de datos y conectividad empresarial:** La *Live Data Connection* consulta S/4HANA o BW/4HANA en tiempo real sin replicación. Para fuentes externas se apoya en *SAP Datasphere*; Gartner mantiene la conectividad no SAP como punto de fricción.

**IBP y colaboración interfuncional:** La integración con *SAP IBP* aporta madurez en supply chain, aunque requiere configuración específica y no resulta transparente para todos los perfiles funcionales.

### 2.4.4 Oracle Cloud EPM Planning

Oracle se sitúa en el cuadrante de Líderes con una puntuación especialmente alta en *Completeness of Vision*, lo que refleja una hoja de ruta estratégica que Gartner valora como coherente con las tendencias del mercado (Crowder y Sahoo 2025).

**Motor de cálculo y arquitectura:** Usa *Essbase* híbrido: BSO (*Block Storage Option*) para lógica compleja y ASO (*Aggregate Storage Option*) para consulta masiva. Permite reglas de cálculo sobre modelos grandes con consistencia y concurrencia; el capítulo 4 desarrolla las decisiones de cubo.

**IA embebida, analítica predictiva y GenAI:** Gartner destaca el modelo de IA *all-inclusive*, la simulación Monte Carlo y el roadmap de **AI Agent Studio**. **IPM Insights**, tratado en el capítulo 9, cubre anomalías, sesgo presupuestario, series temporales y modelos PMML. Como cautela, Oracle no incorpora biblioteca de drivers externos certificados, por lo que su integración recae en el equipo técnico.

**Integración de datos y conectividad empresarial:** *EPM Integration Agent* conecta fuentes *on-premise* o cloud con los cubos EPM y aplica transformaciones en tránsito sin ETL externo; el capítulo 6 desarrolla esta capa. Gartner señala dos cautelas: graph database vía API externa y soporte limitado a 20 idiomas.

**IBP y colaboración interfuncional:** Planning dispone de módulos para personal, proyectos, ventas y operaciones. Las dimensiones *Línea de Negocio*, *Canal* y *custom* del capítulo 4 materializan esa conexión entre plan financiero y planes operativos.

### 2.4.5 Tabla comparativa resumida

La tabla 2.1 resume el posicionamiento de las cuatro plataformas según los ejes de análisis definidos. La valoración refleja las fortalezas y cautelas recogidas en el informe Gartner (Crowder y Sahoo 2025) y el análisis de este capítulo.

## 2.5 Justificación de Oracle Cloud EPM como plataforma base

El análisis anterior no identifica una plataforma universalmente superior, sino perfiles de adecuación. Para este TFM se priorizan dos criterios: posicionamiento independiente de analistas y madurez del ecosistema. Bajo ambos, **Oracle Cloud EPM** se adopta como plataforma base.

**Posicionamiento de mercado según Gartner.** En el Cuadrante Mágico de diciembre de 2025, Oracle figura entre los *Líderes* y obtiene una puntuación alta en *Completeness of Vision* (Crowder y Sahoo 2025). Esto no sustituye el análisis técnico,

Eje	OneStream	Anaplan	SAP Analytics Cloud	Oracle Cloud EPM
<b>Motor de cálculo</b>	<i>Extensible Dimensionality</i> . Cubos Padre-Hijo. Consolidación en memoria con procesamiento selectivo.	<i>Hyperblock</i> . Propagación instantánea en celdas vinculadas. Fuerte en <i>What-if</i> , con limitaciones en consolidación compleja.	<i>SAP HANA Cloud</i> . Arquitectura columnar en memoria. Modelo único para analítica y planificación.	<i>Essbase Híbrido</i> . BSO para lógica compleja y ASO para consultas masivas. Referencia en entornos corporativos de gran escala.
<b>IA, predictiva y GenAI</b>	<i>Sensible AI</i> : series temporales, detección de sesgos, XAI y narrativas generativas.	<i>CoPlanner</i> (GenAI) y predictiva para demanda. Sin biblioteca de drivers externos certificados.	<i>Smart Predict</i> nativo y copiloto <i>Joule</i> . IA dependiente del ecosistema S/4HANA.	<b>IPM Insights</b> (anomalías, sesgo, PMML), Monte Carlo, <i>IA all-inclusive</i> . Sin biblioteca de drivers externos.
<b>Integración de datos</b>	Híbrida: conectores ERP nativos y consultas directas a Data Lakes.	Abierta: APIs REST y <i>Anaplan Connect</i> . Conectividad cloud mejorada pero certificación limitada.	<i>Live Data Connection</i> (tiempo real sobre S/4HANA). <i>SAP Datasphere</i> para fuentes externas.	<i>EPM Integration Agent</i> con transformaciones en tránsito. Graph database vía API externa.
<b>IBP y colaboración</b>	Módulos de personal y proyectos. Propagación dimensional madura. Cobertura de supply chain limitada.	Referencia en <i>xP&amp;A</i> . IBP maduro para RRHH, supply chain y marketing.	<i>SAP IBP</i> para supply chain maduro, pero integración con SAC requiere configuración específica.	Módulos pre-construidos (RRHH, proyectos, ventas). Dimensiones custom para planificación conectada.

**Tabla 2.1:** Tabla comparativa de las principales plataformas de planificación financiera bajo los ejes Gartner 2025. Elaboración propia a partir de Crowder y Sahoo 2025.

pero valida que su hoja de ruta —IA embebida, conectividad cloud y planificación integrada— coincide con las tendencias decisivas del mercado.

**Ecosistema EPM maduro y modular.** Oracle Cloud EPM es una suite modular que comparte infraestructura, motor de cálculo e integración: Planning, Financial Consolidation and Close, Enterprise Data Management, Narrative Reporting, impuestos, rentabilidad y conciliación (Oracle Corporation 2026j). Frente a competidores fuertes en frentes concretos, Oracle combina amplitud de suite y madurez de *Essbase*. Este TFM no despliega toda la suite; usa los módulos alineados con planificación y reporting y deja el resto como escalabilidad futura.

**Enfoque en Planning y Narrative Reporting.** El núcleo elegido es **Oracle Cloud EPM Planning**: presupuestación, *Forecast*, proyección L/P, escenarios *What-if*, Calc Scripts, Groovy, formularios, listas de tareas y control de versiones sobre Essbase híbrido. IPM Insights aporta IA nativa sin plataforma analítica externa. **Narrative Reporting** cubre paquetes ejecutivos con *Doclets* enlazados al dato de Planning, como se desarrolla en el capítulo 11. La combinación Planning + Narrative Reporting concentra el alcance real del TFM: planificación y reporting de gestión.

Las cautelas que Gartner señala para Oracle —ausencia de una biblioteca de drivers externos certificados y soporte limitado a 20 idiomas— se tienen presentes, pero no condicionan el diseño de este TFM, cuyo foco es demostrar una arquitectura de referencia sobre el módulo de planificación y su reporting asociado.

A partir de esta decisión, el resto del documento desarrolla la arquitectura sobre Oracle Cloud EPM Planning y Narrative Reporting: modelo multidimensional, metadatos, integración, proceso operativo, lógica de negocio, analítica predictiva, seguridad y reporting ejecutivo.

## Capítulo 3

# Metodología de Investigación Basada en el Diseño

---

El presente TFM se formula metodológicamente como una **Investigación Basada en el Diseño** (*Design Science Research, DSR*). Esta elección es coherente con la naturaleza del trabajo: no se pretende contrastar estadísticamente una hipótesis sobre el comportamiento de los usuarios, sino diseñar, justificar y evaluar la utilidad de un artefacto de sistemas de información. El artefacto es la arquitectura de presupuestación y reporting corporativo construida sobre Oracle Cloud EPM Planning y Narrative Reporting, cuya plataforma base queda justificada en el apartado 2.5 del capítulo 2.

La DSR resulta especialmente adecuada para proyectos de Big Data y sistemas de información empresariales porque sitúa el conocimiento científico en la creación de artefactos útiles, evaluables y fundamentados en conocimiento técnico previo (Hevner et al. 2004). En este caso, la contribución no es una instalación de producto ni una recopilación de funcionalidades de Oracle, sino una arquitectura argumentada que transforma un problema organizativo —planificación financiera fragmentada, débil trazabilidad de versiones y reporting con elevada carga manual— en un conjunto de decisiones de diseño conectadas entre sí. Siguiendo el esquema de proceso propuesto por Peffers et al. (2007), el trabajo parte de la identificación del problema, deriva objetivos de solución, diseña y desarrolla el artefacto, lo demuestra mediante componentes funcionales de la arquitectura y lo evalúa con criterios de coherencia, trazabilidad, automatización y gobierno del dato.

### 3.1 Enfoque metodológico

---

El hilo conductor del diseño no es la enumeración de módulos, sino la relación de dependencia entre capas de la arquitectura. Cada decisión posterior solo es válida si consume de forma consistente las restricciones fijadas por la decisión anterior. Por ello, la metodología se organiza como una cadena de construcción del artefacto: primero se define la estructura semántica y computacional del dato; después se gobierna el tránsito del dato por estados de madurez; finalmente se

automatizan cálculos, validaciones y promociones sobre intersecciones ya definidas y controladas.

Desde una perspectiva DSR, el problema de investigación puede formularse así: cómo diseñar una arquitectura EPM que permita a una organización multientidad planificar, integrar, validar, automatizar y reportar información financiera sin generar versiones paralelas del dato ni dependencias manuales difíciles de auditar. La respuesta no se busca en una única herramienta, sino en un artefacto compuesto por decisiones de modelado, integración, control, automatización, seguridad y reporting. Su validez se apoya en la utilidad práctica de la arquitectura y en la consistencia interna entre sus componentes.

### 3.1.1 Criterio de documentación

Planning, Data Integration y el resto de módulos de la suite disponen de manuales extensos (Oracle Corporation 2026j; Oracle Corporation 2025a). Este TFM **no reproduce** esa documentación de producto. Solo se describen capacidades estándar de Oracle cuando son necesarias para justificar una decisión del artefacto: dimensiones, versiones, integraciones, listas de tareas, nomenclatura, seguridad por filial y automatización de reglas. El foco metodológico recae en por qué esas piezas se ordenan de ese modo y qué problema de diseño resuelve cada una dentro de la arquitectura propuesta.

Los fragmentos de código, las tablas y las figuras se indexan en el material preliminar, y el código completo se concentra en el anexo A. Esta separación responde también al paradigma DSR: el cuerpo del trabajo razona el diseño del artefacto y el anexo conserva la evidencia técnica de implementación. Cuando un concepto es estándar de plataforma, como el POV de un formulario o los roles *User/Viewer*, se define de forma breve y se cita la documentación; el desarrollo se reserva a lo que diferencia a la arquitectura respecto a una implantación genérica.

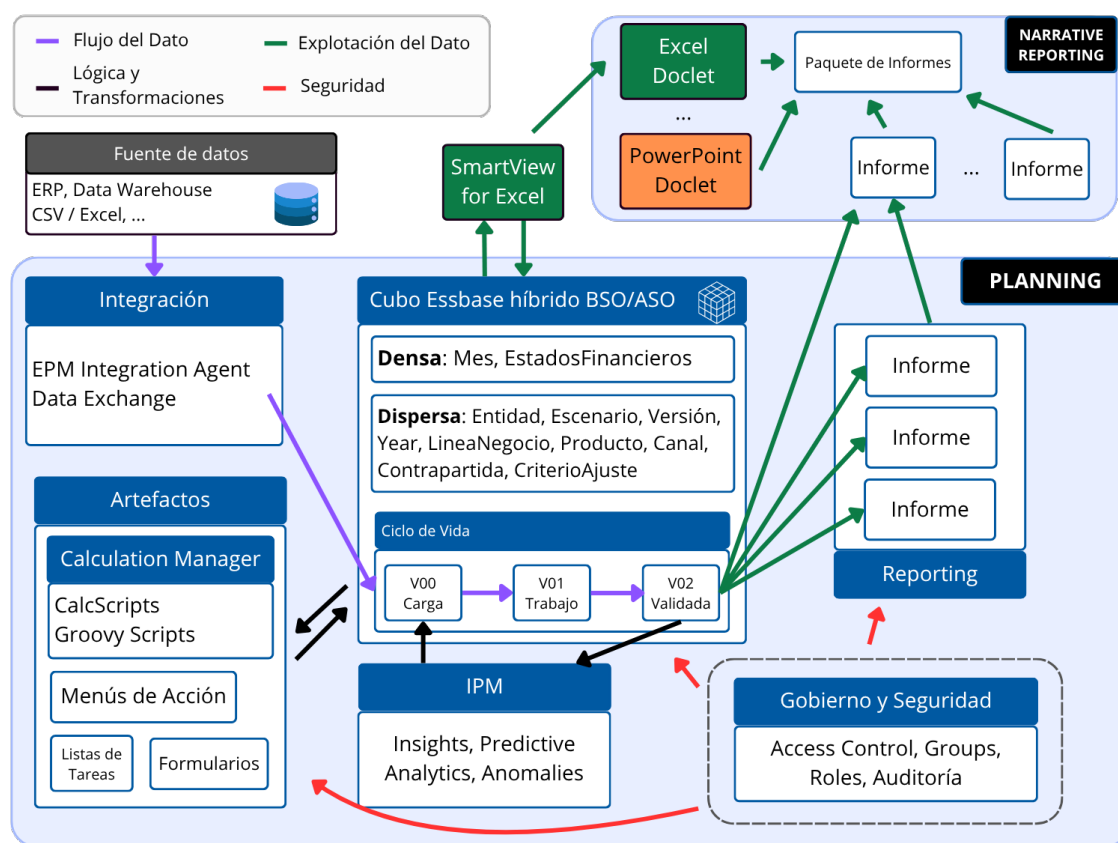
### 3.1.2 Uso asistido de inteligencia artificial generativa

Durante la elaboración del TFM se utilizaron modelos de inteligencia artificial generativa como apoyo instrumental para validar la estructura argumental, contrastar la coherencia entre capítulos, sintetizar borradores técnicos y asistir en la redacción académica del documento. En concreto, se emplearon ChatGPT 5.5, Claude Sonnet 4.6, Gemini 3.5 Flash y Composer 2.5 (OpenAI 2026; Anthropic 2026; Google DeepMind 2026; Anysphere 2026). Su uso no sustituyó la revisión del autor ni la validación técnica frente a documentación oficial: las decisiones de arquitectura, las referencias bibliográficas, las formulaciones finales y la selección de contenidos permanecen bajo responsabilidad del autor.

### 3.2 Secuencia de construcción del artefacto

La arquitectura se construye mediante una secuencia de decisiones acopladas. El Modelo de Datos Multidimensional constituye la base del artefacto porque determina qué puede existir como dato válido. El Ciclo de Vida del Dato actúa después como capa de control porque regula cuándo ese dato puede pasar de carga a ajuste y de ajuste a información oficial. La Orquestación de Lógica Avanzada aparece en tercer lugar porque sus reglas no tienen sentido de forma aislada: consumen la estructura dimensional y el estado de madurez del dato para ejecutar cálculos, validaciones y promociones con un alcance preciso.

La figura 3.1 sintetiza el artefacto diseñado: las fuentes alimentan Planning mediante la capa de integración; el cubo Essbase híbrido organiza el dato y su ciclo V00-V01-V02; los artefactos de Planning ejecutan la operación; IPM Insights consume histórico validado y propone predicciones; y Narrative Reporting publica el dato aprobado bajo una capa transversal de gobierno y seguridad.



**Figura 3.1:** Arquitectura global propuesta para la presupuestación y el reporting corporativo sobre Oracle Cloud EPM.

Fuente: Elaboración propia.

### 3.2.1 El modelo multidimensional como base del artefacto

El capítulo 4 constituye el núcleo metodológico del diseño porque define la ontología operativa de la arquitectura. En un sistema multidimensional, una cifra financiera no existe como valor aislado: existe en una intersección concreta de cuenta, entidad, período, año, escenario, versión y dimensiones analíticas. Por tanto, antes de integrar datos, calcular KPIs o construir informes, el proyecto debe fijar qué significa cada intersección, qué granularidad acepta, cómo se agrega y bajo qué convenciones permanece estable en el tiempo.

Esta decisión es indispensable por tres razones. En primer lugar, el modelo dimensional delimita el lenguaje común entre finanzas, sistemas y reporting. La dimensión *EstadosFinancieros* fija la semántica contable y de gestión; *Entidad* traduce la estructura societaria; *Mes* y *Year* determinan la lectura temporal; *Escenario* separa real, forecast y proyección; *Versión* incorpora madurez del dato; y las dimensiones personalizadas permiten capturar granularidad operativa sin romper la estructura contable. Sin esa gramática compartida, las capas posteriores podrían mover o calcular cifras, pero no garantizar que esas cifras significaran lo mismo para todos los consumidores.

En segundo lugar, el modelo multidimensional define las condiciones de posibilidad del rendimiento y de la automatización. La elección del modo híbrido Essbase, la clasificación *dense/sparse*, la política de *Dynamic Calc*, los miembros compartidos y las convenciones de nomenclatura no son detalles técnicos periféricos: determinan qué operaciones serán viables, qué agregaciones pueden resolverse en consulta, qué celdas admitirán escritura y qué miembros podrán ser referenciados de forma estable por reglas, formularios e integraciones. La arquitectura no puede orquestrar lógica avanzada sobre un espacio dimensional ambiguo.

En tercer lugar, la dimensión *Versión* introduce dentro del propio modelo la futura lógica de control. Al diseñar V00, V01 y V02 como estados diferenciados, el cubo deja de ser un repositorio plano y se convierte en un sistema de transición controlada. Esta decisión conecta directamente el capítulo 4 con el capítulo 6: el ciclo de vida no se añade después como procedimiento externo, sino que queda habilitado por la estructura misma del modelo.

### 3.2.2 El ciclo de vida del dato como capa de control

Una vez definido el espacio donde el dato puede existir, la metodología introduce la capa de control. Esta capa tiene dos funciones complementarias. La primera es proteger la estabilidad del modelo mediante gobierno de metadatos: altas, bajas, cambios de jerarquía, alias, operadores de consolidación y congelación de estruc-

turas durante ciclos presupuestarios. La segunda es regular el tránsito del dato operativo desde su entrada hasta su publicación como dato oficial, tal como se desarrolla en el capítulo 6.

El ciclo  $V00 \rightarrow V01 \rightarrow V02$  no debe interpretarse como una simple copia entre versiones. Metodológicamente, funciona como un mecanismo de maduración del dato. En  $V00$ , la arquitectura permite la carga desde ERP, la ejecución de reglas automáticas y la entrada autorizada necesaria para completar proyecciones. En  $V01$ , el dato deja de ser materia prima y pasa a ser objeto de revisión: se habilitan ajustes puntuales, matrices comparativas y controles de coherencia. En  $V02$ , la información se congela como base oficial para reporting y consumo directivo. La secuencia reduce el riesgo de que un dato calculado, corregido o no validado llegue prematuramente a un informe ejecutivo.

La capa de control consume el modelo multidimensional porque todos sus controles se expresan en términos de intersecciones del cubo. El Workbench de Data Integration valida mapeos antes de escribir en  $V00$ ; las matrices de validación comparan versiones dentro del cubo; el *audit trail* registra cambios sobre miembros concretos; y las listas de tareas del proceso operativo obligan a que cada filial avance por el mismo patrón de revisión. La arquitectura, por tanto, no confía la calidad del dato a una revisión informal, sino a una transición verificable entre estados del modelo.

Esta posición intermedia de la capa de control es metodológicamente relevante: no crea la semántica del dato, porque esa semántica procede del modelo dimensional, pero impide que la automatización posterior opere sobre datos inmaduros. La lógica avanzada solo debe calcular, promover o enviar datos cuando el ciclo ha determinado en qué versión se encuentran, qué permisos aplican y qué validaciones han sido superadas.

### 3.2.3 La orquestación avanzada como capa de consumo y ejecución

La lógica avanzada, desarrollada en el capítulo 8, aparece después del modelo y del ciclo de vida porque no constituye una capa fundacional, sino una capa de ejecución. Calc Scripts, Groovy y Rule Sets consumen dos tipos de información heredada: la estructura dimensional del cubo y el estado de madurez del dato. Un FIX solo es correcto si las dimensiones, jerarquías y miembros han sido diseñados previamente; una promoción entre versiones solo es legítima si el ciclo ha definido qué significa pasar de  $V00$  a  $V01$  o de  $V01$  a  $V02$ ; una validación Groovy solo aporta valor si bloquea una transición relevante dentro del proceso.

Desde el punto de vista metodológico, esta capa convierte el artefacto de una

arquitectura estática en una arquitectura operativa. Los Calc Scripts implementan transformaciones masivas dentro del cubo, como copias controladas, borrados previos de destino, cálculos de KPIs o consolidaciones. Groovy incorpora contexto: lee el formulario, identifica entidad, año, escenario y versión, decide qué regla ejecutar y puede impedir acciones incoherentes antes de que afecten al dato. Los Rule Sets encadenan secuencias repetibles cuando no existen ramas de decisión. La aportación científica no reside en afirmar que Oracle permite escribir reglas, sino en justificar qué tipo de regla corresponde a cada decisión de diseño y en qué punto del ciclo debe ejecutarse.

La dependencia con los bloques anteriores es estricta. Si el modelo dimensional no hubiera definido entidades, escenarios, versiones y granularidad temporal, la lógica no podría acotar su alcance sin riesgo de afectar combinaciones incorrectas. Si el ciclo de vida no hubiera separado carga, ajuste y validación, la misma regla de copia sería técnicamente posible pero metodológicamente insegura, porque no habría criterio para saber cuándo ejecutarla ni qué estado produce. Por eso, la orquestación no se presenta como una colección de scripts, sino como el mecanismo que operacionaliza la arquitectura diseñada.

### 3.3 Demostración y evaluación del diseño

En DSR, un artefacto debe demostrarse y evaluarse respecto al problema que pretende resolver. En este TFM, la demostración se distribuye a lo largo de los capítulos técnicos: el capítulo 4 demuestra la viabilidad estructural del cubo; los capítulos 5 y 6 demuestran la trazabilidad de metadatos, integraciones y versiones; el capítulo 7 muestra cómo el diseño se traduce en formularios, listas de tareas y flujos por entidad; y el capítulo 8 demuestra cómo la arquitectura se automatiza mediante reglas parametrizadas.

La evaluación se plantea sobre criterios de diseño, no sobre métricas de despliegue en producción que exceden el alcance del TFM. El primer criterio es la **coherencia semántica**: toda cifra debe poder ubicarse en una intersección dimensional inequívoca. El segundo es la **trazabilidad**: debe ser posible reconstruir desde qué fuente, versión, entidad y ciclo procede un dato usado en reporting. El tercero es la **controlabilidad**: ningún dato debería llegar a V02 sin pasar por los controles definidos. El cuarto es la **automatización gobernada**: las reglas deben reducir trabajo manual sin ocultar el estado del dato ni saltarse validaciones. El quinto es la **escalabilidad organizativa**: el mismo patrón debe poder replicarse por filial, escenario y período sin rediseñar la arquitectura.

Con estos criterios, el capítulo metodológico cumple una función de tribunal:

establece cómo debe leerse el resto del documento. Los capítulos posteriores no son piezas independientes, sino iteraciones de desarrollo del mismo artefacto DSR. El Modelo de Datos Multidimensional fija la base de significado y cálculo; el Ciclo de Vida del Dato introduce control y madurez; la Orquestación de Lógica Avanzada ejecuta reglas sobre esa base ya gobernada; y las capas posteriores consumen el dato resultante sin romper la cadena de trazabilidad. La analítica predictiva solo puede interpretar series temporales consistentes; la seguridad solo puede restringir entidades, versiones e intersecciones previamente definidas; y Narrative Reporting solo debe comunicar información validada, normalmente anclada a V02, para que la narrativa ejecutiva conserve continuidad con la intersección original del cubo.

## Capítulo 4

# Diseño del Modelo de Datos Multidimensional

---

El modelo multidimensional es la decisión estructural de la arquitectura: fija la semántica de cada dato, el espacio de cálculo de Essbase, la granularidad de entrada, las jerarquías de agregación y las referencias que consumirán integraciones, formularios, reglas Calc, Groovy, seguridad y reporting. El capítulo documenta solo decisiones de diseño; la operativa de producto se cita cuando condiciona la arquitectura.

## 4.1 Selección del tipo de cubo: BSO, ASO e Híbrido

---

Oracle Cloud EPM Planning permite cubos BSO, ASO e híbridos. La elección afecta a escritura en formularios, disponibilidad de Calc Script, agregación, tamaño en disco y concurrencia (Oracle Corporation 2025e).

### 4.1.1 Block Storage Option (BSO)

BSO organiza el dato en bloques densos. Es imprescindible cuando el modelo requiere:

- **Escritura bidireccional:** entrada de usuario en formularios y escritura masiva por reglas.
- **Calc Script completo:** asignaciones, bucles, condiciones y referencias cruzadas entre dimensiones.
- **Control de bloque:** útil para planificación, aunque sensible a explosión de *sparse blocks*.

### 4.1.2 Aggregate Storage Option (ASO)

ASO prioriza consulta agregada: calcula agregaciones sobre datos atómicos sin precalcular valores, reduce disco y escala bien en reporting masivo. No es adecuado como cubo principal de planificación porque no permite escritura directa en celdas ni la mayoría de Calc Scripts.

### 4.1.3 Modo Híbrido: decisión adoptada

La arquitectura adopta **Essbase híbrido**: escritura y lógica BSO en nivel hoja; agregación dinámica tipo ASO en dimensiones dispersas de alta cardinalidad (Oracle Corporation 2025c). Requisitos simultáneos:

1. formularios editables por usuarios financieros;
2. vistas agregadas multidimensionales con respuesta objetivo inferior a dos segundos;
3. reglas de negocio complejas mediante Calc Script.

En modo híbrido, los nodos dispersos de consolidación se configuran como *Dynamic Calc*; las celdas de entrada se mantienen en bloques BSO. Si el volumen de reporting exige separación física, puede añadirse cubo ASO de consulta vía *Smart Push*, sin alterar el modelo lógico (Oracle Corporation 2025c).

Característica	BSO puro	ASO puro	Híbrido adoptado
Escritura de usuarios	Sí	No	Sí
Calc Scripts complejos	Sí	Limitado	Sí
Consulta agregada	Media	Alta	Alta
Espacio en disco	Alto	Bajo	Medio
Adecuación a planificación	Alta	Baja	Alta

**Tabla 4.1:** Comparativa de tipos de almacenamiento Essbase para planificación.

## 4.2 Diseño de las dimensiones

Planning exige *Account, Entity, Period, Scenario, Version* y *Year* (Oracle Corporation 2025b). La arquitectura añade dimensiones personalizadas y mantiene **once dimensiones activas** —*Divisa* solo si hay perímetro multidivisa—, dentro de la recomendación de no superar 12 dimensiones por cubo (Oracle Corporation 2024).

### 4.2.1 Dimensión Mes (Period)

*Period*, renombrada **Mes**, define la granularidad temporal **dentro** del ejercicio fijado por *Year* (apartado 4.2.4). Se diseña con doble lectura:

- **MTD**: nodo MTD; hijos M01m–M12m; valor mensual sin arrastre. Uso: carga ERP mensual y reporting mensual.
- **YTD**: nodo YTD; hijos M01–M12; acumulado desde enero. Uso: control presupuestario y estados contables.

- **BegBalance:** valor anual no mensualizado; saldos, deuda, inventario o proyecciones plurianuales a nivel anual (apartado 4.2.2).

#### Elección de la dimensión de almacenamiento.

- **Origen mensualizado:** MTD almacenado; YTD Dynamic Calc por suma de MTD previos.
- **Origen acumulado:** YTD almacenado; MTD Dynamic Calc por diferencia, p. ej.  $M03m = M03 - M02$ .

La jerarquía no almacenada se calcula dinámicamente para evitar duplicación temporal en disco.

### 4.2.2 Dimensión Escenario

*Scenario*, renombrada **Escenario**, separa realidades financieras y habilita comparabilidad entre ciclos.

#### Escenario Real

Real es la fuente histórica cargada desde ERP:

**Real Analítico:** métricas internas no normativas: margen por producto, retención, coste de adquisición por canal.

**Real Contable:** P&G, balance y flujos con vistas normativas; base de reporting legal y regulatorio.

#### Escenarios de Forecast (FCST01–FCST12)

Se crean doce escenarios FCST01–FCST12. Cada  $FCST_n$  conserva la fotografía del forecast elaborada en el cierre  $n$ :

- meses 1 a  $n - 1$ : dato real o validado;
- meses  $n$  a 12: proyección;
- FCST01: doce meses proyectados;
- FCST06: enero–mayo cerrado, junio–diciembre proyectado;
- FCST12: enero–noviembre cerrado, diciembre proyectado.

No se sobrescribe un único forecast: se conserva trazabilidad y evaluación retrospectiva de precisión. El *Rolling Forecast* se ejecuta en V00; el planificador completa meses abiertos en V00 y usa V01 solo para ajustes tras promoción (apartado 4.2.3).

## Escenario Proyección L/P

La proyección L/P cubre 3–5 años con granularidad mixta:

- **Año 1:** mensual, usando MTD/YTD.
- **Años 2+:** anual, usando BegBalance; evita falsa precisión mensual en horizonte estratégico.

**Ejemplo: Proyección Trianual.** El patrón PRTRnn\_mm identifica ventanas trianuales:

- PRTR: tipo de escenario;
- nn: año inicial;
- mm: año final.

PRTR26\_28 cubre 2026–2028: 2026 mensual; 2027–2028 anual en BegBalance. Cada ciclo nuevo (PRTR27\_29, PRTR28\_30, etc.) se crea sin alterar proyecciones históricas validadas.

### 4.2.3 Dimensión Versión

*Version* modela madurez del dato y habilita el control V00–V01–V02; permisos y proceso operativo en el capítulo 6 (tabla 6.3).

**V00 – Carga:** integración, reglas automáticas, copias desde otros escenarios y entrada manual autorizada para completar proyecciones. Aquí corren KPIs, *Input Balance*, *Rolling Forecast* y lógica de proyección FCST01–FCST12 (apartado 4.2.2).

**V01 – Trabajo:** copia de V00; solo ajustes puntuales tras revisión. Si no hay corrección, replica V00.

**V02 – Validada:** dato oficial e inmutable para reporting y Narrative Reporting. Cualquier corrección vuelve a V01 y repite promoción.

**V03 – Enviada (opcional):** fotografía del último envío a grupo en holdings multi-nivel.

### 4.2.4 Dimensión Year (Año)

*Year* identifica ejercicio fiscal; *Mes* desglosa la granularidad interna (apartado 4.2.1). Diseño: miembros hoja independientes bajo raíz Oracle, sin jerarquía aditiva; raíz con operador ~ para evitar acumulados entre ejercicios.

**Nomenclatura.** FYXX: FY25 = 2025, FY26 = 2026, FY27 = 2027. El alias visible es el año («2025»), no el identificador técnico FY25 (apartado 4.4.3).

**Ventana temporal del cubo.** Se mantienen años históricos, año activo y horizonte presupuesto/L/P. Nuevos ejercicios se añaden por gobierno de metadatos (capítulo 5); años obsoletos se archivan preferentemente en ASO histórico, no se eliminan sin trazabilidad.

#### 4.2.5 Dimensión Entidad

*Entity* implementa el perímetro societario:  **Holding** →  **Filial**. El holding consolida estados financieros y eliminaciones intercompañía; cada filial es nodo hoja de carga ERP, seguridad y listas de tareas.

**Nomenclatura.** Filiales con patrón EXXXX: secuencia E0001, E0002, ..., o código corporativo equivalente si mantiene prefijo E y longitud fija (p. ej., E0123). La razón social reside en alias, no en el nombre técnico (apartado 4.4.3). Altas, bajas y reorganizaciones pasan por gobierno de metadatos (capítulo 5).

#### 4.2.6 Dimensión Estados Financieros (Account)

*Account*, renombrada  **EstadosFinancieros**, concentra plan contable, métricas, ratios y magnitudes de planificación/reporting. Bloques funcionales:

**Cuentas internas societarias:** P&G analítica, balance y flujos para control de gestión.

**Métricas y KPIs:** EBITDA, margen bruto, variación interanual, ratios; nombre descriptivo, no patrón contable (apartado 4.4.5).

**Cuentas normativas:** PGC, IFRS, CNMV, PCEA u otros marcos jurisdiccionales.

##### Estrategias de implementación de múltiples normativas

**Opción A: Jerarquías independientes.** Cada normativa mantiene miembros propios. Ventajas: implementación simple y flexibilidad por marco. Costes: duplicación de dato, integración triple y mantenimiento repetido de cambios en plan de cuentas.

**Opción B: Jerarquías alternativas con nodos hoja compartidos (solución adoptada).** Una jerarquía primaria almacena el hecho; PGC/IFRS/gestión se modelan como jerarquías alternativas con Shared sobre los mismos nodos hoja. Resultado:

- fuente única de verdad: una carga, múltiples vistas;

- coherencia automática por agregación `Dynamic Calc`;
- código `XXXX_YYYY`: prefijo normativo/societario + epígrafe;
- mayor exigencia de mapeo inicial entre gestión y normativa.

Se adopta B por gobierno del dato y eliminación de inconsistencias.

#### 4.2.7 Dimensión Divisa (Currency) — opcional

*Currency* solo se activa si el grupo opera en más de una moneda (Oracle Corporation 2025b). Si todas las filiales comparten divisa funcional y reporting, se omite. Si aplica: lista plana de códigos ISO (EUR, GBP, USD, CHF, BRL, MXN, ...), sin jerarquías ni subtotales. Planning asocia moneda funcional a filial y usa tablas estándar de tipo de cambio para consolidar en moneda holding.

#### 4.2.8 Dimensiones personalizadas

Las *custom dimensions* añaden granularidad operativa y rentabilidad no cubierta por dimensiones estándar. Planning admite hasta 25 dimensiones personalizadas (Oracle Corporation 2026j); las convenciones de raíz, `_DF`, `_NA` y prefijos se fijan en la sección 4.4.

##### Línea de Negocio

Segmenta ingresos/costes por unidad estratégica, separada de la estructura legal *Entity*. Estructura: totalizador, líneas operativas, miembro de carga sin asignar y miembro no aplicable.

##### Producto / Familia de Producto

Rentabilidad por gama o servicio. Jerarquía: *Categoría* → *Familia* → *Referencia*. Entrada en referencia; agregación automática; miembro de reserva para referencias no clasificadas.

##### Canal de Distribución

Segrega ventas y costes por venta directa, online, distribuidores, grandes cuentas, etc. Soporta análisis de *mix de canal*. Estructura análoga a línea de negocio: totalizador, operativos, carga incompleta y no aplicable.

##### Contrapartida Intercompañía

Clave para eliminar transacciones intra-grupo. Cada operación entre filiales registra la entidad de contrapartida; existe un miembro por filial alineado con *Entidad* y un miembro sin contrapartida. Patrón en tabla 4.5.

## Criterio de Ajuste

Dimensión de trazabilidad: separa Carga (integración automática en V00) y Ajuste (correcciones manuales en V01); totalizador Dynamic Calc con operador +; miembro no aplicable en tabla 4.5. La entrada manual para completar forecast se escribe en V00, no en Ajuste. Solo importes; no tasas.

## 4.3 Configuración técnica del modelo

Reglas transversales del outline: jerarquías, generaciones/niveles, miembros compartidos, operadores, almacenamiento y clasificación dense/sparse.

### 4.3.1 Estructura de jerarquías: generaciones, niveles y miembros compartidos

Essbase define **generaciones** desde raíz (generación 1) y **niveles** desde hojas (nivel 0) (Oracle Corporation 2025f). Los Calc Scripts seleccionan hojas con @RELATIVE(miembro, 0).

#### Nodos padre (consolidación)

No reciben entrada; se configuran como Dynamic Calc para calcular en consulta sin persistir bloque (Oracle Corporation 2025g; Oracle Corporation 2025c). Totales/subtotales usan operador de consolidación; ratios añaden *member formula*. Ejemplos: TOTAL\_VENTAS, trimestres de *Mes*, margen por ciento, ROE y variación interanual.

#### Nodos hoja (entrada de datos)

Nivel 0 de carga y edición. Política:

- almacenamiento Never Share, no Store: evita *implied share*, bloqueos en formularios y ambigüedad padre/hijo (Oracle Corporation 2025g);
- en dimensiones dispersas no usar Dynamic Calc sin fórmula, por degradación de formularios e informes (Oracle Corporation 2025c);
- escritura por *User/Power User* según capítulo 10.

#### Miembros compartidos (*Shared Members*)

Las jerarquías alternativas de *EstadosFinancieros* (apartado 4.2.6) reutilizan hojas base con Shared (Oracle Corporation 2025g; Oracle Corporation 2025h):

- prototipo en jerarquía primaria, Never Share, nivel 0, antes de sus referencias;
- referencia Shared: sin bloque propio; apunta al valor base;

- nombre idéntico al base; sin hijos, UDAs ni fórmulas propias (Oracle Corporation 2025j; Oracle Corporation 2025g);
- compartición explícita; Never Share evita *implied share*.

### 4.3.2 Operadores de agregación

El operador de consolidación determina contribución al padre; por defecto +, ajustado según naturaleza contable (Oracle Corporation 2025h).

Operador	Not.	Uso en la arquitectura
Suma	+	Ingresos, gastos y saldos aditivos.
Resta	-	Gastos que reducen resultado; Essbase multiplica por -1 y suma.
Multiplicación	*	Hipótesis, tasas de crecimiento, multiplicadores.
División	/	Ratios que dividen entre acumulado de hermanos previos.
Porcentaje	%	División con resultado multiplicado por 100.
Ignorar	~	Saldos o miembros que no consolidan en su padre; usado en <i>Year</i> y <i>Scenario</i> .
No consolidar global	^	Auxiliares excluidos de toda consolidación: tipos de cambio, tasas de referencia.

**Tabla 4.2:** Operadores de consolidación Essbase y uso en la arquitectura (Oracle Corporation 2025h).

~ excluye al miembro de la consolidación de su propia dimensión, pero permite consolidaciones en otras dimensiones; ^ lo excluye globalmente.

### 4.3.3 Propiedades de almacenamiento de miembros

La tabla fija la política local sobre las opciones estándar de Essbase (Oracle Corporation 2025g).

Propiedad	Aplicación	Criterio
Never Share	Hojas nivel 0	Entrada/integración; evita <i>implied share</i> .
Dynamic Calc	Padres	Agregación automática; ratios con <i>member formula</i> .
Label Only	Cabeceras	Sin dato ni bloque; navegación/secciones.
Shared	Jerarquías alternativas	Misma cuenta en PGC/IFRS/gestión; un bloque de datos.
Store	Padres muy consultados	Excepción: precálculo por Calc Script.
Dyn Calc and Store	Fórmulas costosas	Calcula primera vez y cachea.

**Tabla 4.3:** Política de almacenamiento por tipo de miembro.

**Aplicación de la política.** Hojas de carga/edición: Never Share; padres: Dynamic Calc; cabeceras «P&G», «Balance» y «Flujos»: Label Only; jerarquías PGC/IFRS/gestión: Shared sobre miembro base (apartado 4.3.1). Esta combinación evita duplicar bloques y bloqueos en formularios (Oracle Corporation 2025g; Oracle Corporation 2025b).

#### 4.3.4 Clasificación Dense/Sparse de las dimensiones

En BSO solo **Mes** y **EstadosFinancieros** son densas; todas las demás son dispersas, siguiendo práctica Oracle para planificación financiera (Oracle Corporation 2025d). La decisión concentra cálculo financiero en bloques compactos y evita materializar combinaciones vacías de filiales/dimensiones analíticas.

## 4.4 Política de nomenclatura y convenciones

La nomenclatura separa identificador técnico y etiqueta de presentación. Oracle impone límites de longitud, caracteres prohibidos, palabras reservadas y reglas para miembros compartidos (Oracle Corporation 2025j).

#### 4.4.1 Restricciones técnicas de Oracle

El diseño cumple restricciones Oracle mediante identificadores cortos, sin espacios ni caracteres especiales, con guion bajo como separador único y prefijos por dimensión.

#### 4.4.2 Convenciones generales de la arquitectura

- **Sin caracteres especiales:** sin espacios, tildes, comas, puntos ni barras; separador -.
- **Raíz xxx\_TOT:** dimensiones aditivas con nodo ABR\_TOT, Dynamic Calc, operador +. Excepciones: *EstadosFinancieros* (sin total único P&G/Balance/Flujos), *Entidad* (holding propio), *Mes*, *Escenario* y *Year* (raíces semánticas del sistema).
- **Miembro por defecto xxx\_DF:** primer hijo de xxx\_TOT; recibe datos sin categoría en carga; debe quedar vacío en explotación normal.
- **No aplica xxx\_NA:** hermano de xxx\_TOT; operador ~; para hechos donde la dimensión no aplica.
- **Jerarquías auxiliares xxx\_AUX:** nodos de vistas alternativas, p. ej. PGC\_AUX, IFRS\_AUX, GEST\_AUX; reutilizan hojas mediante miembros compartidos (apartado 4.3.1).

- **Prefijos hoja:** LIN\_, PRO\_, CAN\_, ICP\_, CRIT\_. *Contrapartida* usa ICP\_ por convenio intercompañía Oracle.

### 4.4.3 Alias de miembro: nombre técnico y etiqueta de presentación

Planning permite alias por miembro (Oracle Corporation 2026j). Política:

- **Nombre de miembro:** identificador técnico estable para integraciones, Calc Scripts, mapeos y variables. Cambiarlo exige actualizar artefactos. Ejemplo: PGC\_A\_1B2.
- **Alias:** etiqueta de negocio visible en formularios, ad-hoc, dashboards, Smart View y doclets. Ejemplo: «Amortizaciones por deterioro».

Formularios muestran alias en filas/columnas; POV, reglas e integraciones resuelven por nombre interno. Alias de *Entidad*, *Year*, líneas, canales y productos se cargan en columnas estándar del fichero de dimensión o editor de metadatos (apartado 5.1.1).

### 4.4.4 Plantilla genérica de jerarquía

Dimensiones personalizadas aditivas siguen esta plantilla.

Miembro	Almacenamiento	Op.	Función
xxx_TOT	Dynamic Calc	+	Total consolidado
xxx_DF	Never Share	+	Carga sin categoría
xxx_01_Descripcion	Never Share	+	Hojas operativas
xxx_02_Descripcion	Never Share	+	⋮
xxx_NA	Never Share	~	Fuera de xxx_TOT; no aplica

**Tabla 4.4:** Plantilla genérica de jerarquía para dimensiones personalizadas.

xxx\_TOT y subtotales son Dynamic Calc; xxx\_DF absorbe carga incompleta; xxx\_NA queda fuera del flujo consolidante.

### 4.4.5 Convenciones específicas por dimensión

**Estados Financieros.** Cuentas contables: XXXX\_YYYY. XXXX identifica normativa/ámbito (PGC, IFRS, SOC, GEST, CNMV, PCEA, etc.); YYYY es epígrafe único de la normativa, sin formato universal. La descripción vive en alias (apartado 4.4.3). Métricas/KPIs usan nombre descriptivo: Margen\_Bruto, EBITDA, Variacion\_Ventas\_YTD, ROE.

<b>Dimensión</b>	<b>Raíz / DF / NA</b>	<b>Patrón hoja</b>	<b>Ejemplo</b>
<i>Estados- Financieros</i>	Sin raíz única / — / EST_NA	XXXX_YYYY; métrica descriptiva	PGC_700_701 Margen_Bruto
<i>Entidad</i>	Holding propio / — / ENT_NA	EXXXX	E0001, E0002
<i>Línea de Nego- cio</i>	LIN_TOT / LIN_DF / LIN_NA	LIN_NN_Desc	LIN_01_Retail
<i>Producto</i>	PRO_TOT / PRO_DF / PRO_NA	PRO_Cat_Ref	PRO_ELEC_ A4501
<i>Canal</i>	CAN_TOT / CAN_DF / CAN_NA	CAN_NN_Desc	CAN_03_Online
<i>Contrapartida</i>	CTP_TOT / — / ICP_NA	ICP_EXXXX	ICP_E0001
<i>CriterioAjuste</i>	CRIT_TOT / — / CRIT_NA	Descriptivo	Carga, Ajuste
<i>Versión</i>	Sin_TOT / — / VER_NA	V + número	V00, V01
<i>Escenario</i>	Sin_TOT / — / ESC_NA	FCSTnn, PRTRnn_mm	FCST06 PRTR26_28
<i>Mes (Period)</i>	Raíz del sistema / — / —	Mnn, Mnnm	M01, M06m
<i>Year</i>	Raíz del sistema / — / —	FYXX	FY25, FY26

**Tabla 4.5:** Raíz, \_DF, \_NA y patrón de hojas por dimensión.

## Capítulo 5

# Gestión y Gobierno de Metadatos en Oracle Cloud EPM Planning

---

En cualquier implementación de EPM, la calidad del dato depende en igual medida de la calidad del dato en sí y de la calidad de los metadatos que lo estructuran. Una jerarquía mal mantenida, un miembro renombrado sin criterio o una cuenta incorporada fuera del procedimiento establecido pueden invalidar meses de trabajo de planificación.

Este capítulo se limita a la **estructura dimensional** del cubo: herramientas de mantenimiento, procedimiento de cambio y congelación en ciclos presupuestarios. El **ciclo de vida del dato** (integración desde el ERP, versiones V00–V02 y controles entre versiones) se documenta en el capítulo 6; la interacción del planificador con formularios y tareas, en el capítulo 7.

## 5.1 Herramientas de mantenimiento en Planning

Planning dispone de editor web, importación por fichero y EPM Automate (Oracle Corporation 2026j). En esta arquitectura, el **canal principal** es la importación por fichero de dimensión (*dimension worksheet*); el editor sirve para correcciones puntuales y Automate para promover cambios entre entornos.

### 5.1.1 Importación de metadatos por fichero

Las dimensiones de mayor cardinalidad (*EstadosFinancieros, Entidad*) se mantienen exportando la dimensión, editando el CSV en Excel según las convenciones del apartado 4.4.2 y reimportando en TEST antes de PRO. El fichero define jerarquía, operadores, almacenamiento y **columnas de alias** (etiquetas de presentación; apartado 4.4.3) en el formato estándar de Oracle (Oracle Corporation 2026j). El nombre de miembro del outline no debe alterarse por un cambio de redacción en pantalla: las correcciones literales del usuario van al alias.

El proceso de importación sigue los pasos:

1. Exportar la dimensión actual como fichero de referencia (*Export Dimension*).

2. Aplicar los cambios sobre el fichero en local.
3. Validar el fichero contra las convenciones de nomenclatura (apartado 4.4).
4. Importar el fichero actualizado (*Import Dimension*) en el entorno de TEST.
5. Verificar el outline resultante y ejecutar una prueba de cálculo.
6. Promover el cambio a PRO siguiendo el procedimiento de cambios descrito en el apartado 5.2.2.

### 5.1.2 Editor puntual y EPM Automate

Cambios aislados (alta de una filial, corrección de alias) pueden hacerse en *Administration >Dimensions*. Para despliegues repetibles, EPM Automate encadena `importMetadata`, `refreshCube` y validaciones en scripts del equipo de TI (Oracle Corporation 2026j).

## 5.2 Gobierno del proceso de cambio

Disponer de las herramientas técnicas no es suficiente si no existe un procedimiento que regule quién puede proponer cambios, quién los valida y cuándo se aplican. La arquitectura propuesta establece las siguientes reglas de gobierno para la gestión de metadatos.

### 5.2.1 Roles y responsabilidades

**Responsable de dimensión (*Dimension Owner*):** Perfil de negocio que solicita el cambio y justifica su necesidad. No tiene acceso de escritura directo sobre el cubo. Ejemplos: el director financiero para cambios en *EstadosFinancieros*, el responsable de organización para cambios en *Entidad*.

**Administrador EPM:** Perfil técnico que valida la solicitud contra las convenciones de nomenclatura, genera el fichero de importación, ejecuta la carga en TEST, verifica el resultado y promueve a PRO.

**Responsable de calidad del dato:** Valida que el cambio estructural no rompe series históricas ni informes en curso (no confundir con la revisión de cifras entre versiones del cubo, descrita en el capítulo 6). Su aprobación es necesaria antes de promover el metadato a PRO.

### 5.2.2 Procedimiento de cambio estructural

Todo cambio en la estructura dimensional sigue el flujo:

1. **Solicitud documentada:** El Responsable de dimensión abre un ticket en el sistema de gestión de tareas del equipo (p. ej., Jira, Azure DevOps) describiendo el cambio, la dimensión afectada y la justificación de negocio.
2. **Preparación técnica:** El Administrador EPM exporta la dimensión actual, aplica el cambio en local siguiendo las convenciones documentadas y revisa la coherencia del fichero resultante.
3. **Prueba en TEST:** Se importa el fichero modificado en el entorno de TEST y se ejecutan pruebas de cálculo y reporting para verificar que ningún informe existente queda roto por el cambio.
4. **Aprobación:** El Responsable de calidad del dato aprueba el cambio en el ticket.
5. **Promoción a PRO:** El Administrador EPM importa el fichero en PRO, refresca el cubo y notifica a los planificadores afectados.
6. **Cierre y archivo:** El fichero de dimensión actualizado se versiona en el repositorio del equipo con el número de ticket y la fecha. El ticket se cierra con la evidencia de la prueba superada.

### 5.3 Congelación de metadatos durante el ciclo presupuestario

El período de elaboración del presupuesto anual es el momento de mayor riesgo para la integridad de los metadatos: si se reorganiza una jerarquía o se renombra un miembro mientras los planificadores están introduciendo datos, las celdas ya completadas pueden quedar huérfanas o apuntar a nodos obsoletos. Para evitar esta situación, la arquitectura establece la práctica de **congelación de metadatos** durante las ventanas activas de planificación.

Durante el período de congelación:

- No se aceptan solicitudes de cambio estructural en las dimensiones activas en el proceso presupuestario (*EstadosFinancieros, Entidad, dimensiones custom*).
- Solo se admiten cambios de emergencia bajo un procedimiento de excepción documentado que requiere aprobación expresa del Responsable de calidad del dato y del director de planificación.
- Los cambios no urgentes se acumulan en una lista de espera (*backlog de metadatos*) que se procesa una vez cerrado el ciclo presupuestario.

Las fechas de apertura y cierre de la congelación se publican en el calendario de planificación al inicio de cada ejercicio y se comunican a todos los responsables de

dimensión.

## Capítulo 6

# Arquitectura de Integración y Ciclo de Vida del Dato

---

El modelo multidimensional solo es útil si cada dato entra con origen, mapeo, versión y estado de validación inequívocos. Este capítulo define la tubería ERP → Data Integration → Essbase y el protocolo que impide que información no revisada llegue al reporting.

## 6.1 Componentes de la arquitectura de integración

---

### 6.1.1 EPM Integration Agent

El **EPM Integration Agent** se instala en red corporativa y extrae datos del ERP mediante SQL con conexión saliente HTTPS (Oracle Corporation 2026j). En esta arquitectura es el canal único para cargar el Real contable en V00.

### 6.1.2 Módulo Data Integration

**Data Integration** (*Data Exchange*) concentra localizaciones por filial, consultas, mapeos ERP → cubo, transformaciones y carga al escenario destino (Oracle Corporation 2026j). La regla se configura una vez y se parametriza por entidad, período y categoría; el planificador la ejecuta desde una tarea URL del capítulo 7.

### 6.1.3 Fuentes de datos

## 6.2 Configuración de una regla de integración

---

Cada carga queda definida por cuatro objetos: localización, formato de importación, mapeos y transformaciones. Esa separación evita duplicar lógica entre filiales y permite relanzar una carga corrigiendo solo el componente fallido.

### 6.2.1 Localizaciones (*Locations*)

La **localización** fija el contexto de ejecución (Oracle Corporation 2026j): origen, aplicación destino, entidad EXXXX, categoría/escenario y mapeo de períodos. Sepa-

Fuente	Conexión	Aporte al cubo
ERP corporativo ( <i>on-premise</i> )	EPM Integration Agent + SQL	Real: P&G, balance y flujos de efectivo
Data Warehouse corporativo	EPM Integration Agent + SQL	Ventas, producción y detalle operativo con granularidad analítica
CSV / Excel	Carga directa vía Data Integration	Hipótesis externas: tipo de cambio, inflación, previsiones macroeconómicas
Sistemas de RRHH	EPM Integration Agent + API REST	Plantilla, costes salariales y contrataciones previstas

**Tabla 6.1:** Fuentes de datos y tipos de conexión en la arquitectura de integración.

rarla de la regla permite usar la misma extracción y transformación en varias filiales que comparten estructura de origen.

## 6.2.2 Formato de importación y consulta SQL

El **Import Format** asigna columnas de origen a dimensiones del cubo: cuenta → *EstadosFinancieros*, centro de coste → *Entidad* o dimensión analítica, moneda → *Divisa* e importe → valor numérico (Oracle Corporation 2026j). En SQL se usan variables propias de Data Integration, distintas de las variables de sustitución de Planning del apartado 7.3.1:

- \$PERIOD: período de extracción.
- \$YEAR: ejercicio fiscal.
- \$ENTITY: entidad activa de la localización.
- \$CATEGORY: escenario/categoría de destino.

La consulta filtra por esas variables y deja el resultado en el Workbench. No se fija un SQL universal en el TFM porque depende del esquema ERP; el estándar arquitectónico es la parametrización, no la sintaxis concreta del origen.

## 6.2.3 Filtros de ejecución

Cada ejecución acota períodos y modo de carga (*Replace* o *Accumulate*) (Oracle Corporation 2026j). La política exige revisar el Workbench y resolver miembros sin mapeo antes de escribir en V00.

## 6.2.4 Mapeos de miembros (*Member Mappings*)

*Multi-Dim* resuelve casos donde una cuenta ERP se clasifica de forma distinta por filial, línea, canal o producto. Los *unmapped members* quedan retenidos y bloquean

Tipo	Uso	Ejemplo
<i>Explicit</i>	Coincidencia exacta del valor de origen	70100000 → PGC_700_701
<i>Like</i>	Patrón con comodín	701* → PGC_700_*
<i>Between</i>	Rango numérico de valores	700000–799999 → PGC_700
<i>In Group</i>	Grupo de valores de origen	VENTAS_NAC → PGC_700_701
<i>Multi-Dim</i>	Regla condicionada por varias dimensiones	Cuenta 7010 + centro CC01 → miembro específico

**Tabla 6.2:** Tipos de mapeo de miembros en Data Integration (Oracle Corporation 2026j).

la carga hasta mapearlos o descartarlos explícitamente.

### 6.2.5 Transformaciones

Las transformaciones estándar son: cambio de signo en gastos (`importe * -1`), derivación de EXXXX desde sociedad ERP y asignación de detalle operativo a dimensiones custom. Si el mapeo nativo no basta, se añaden *Event Scripts* Jython (Oracle Corporation 2026j).

## 6.3 Ejecución y validación

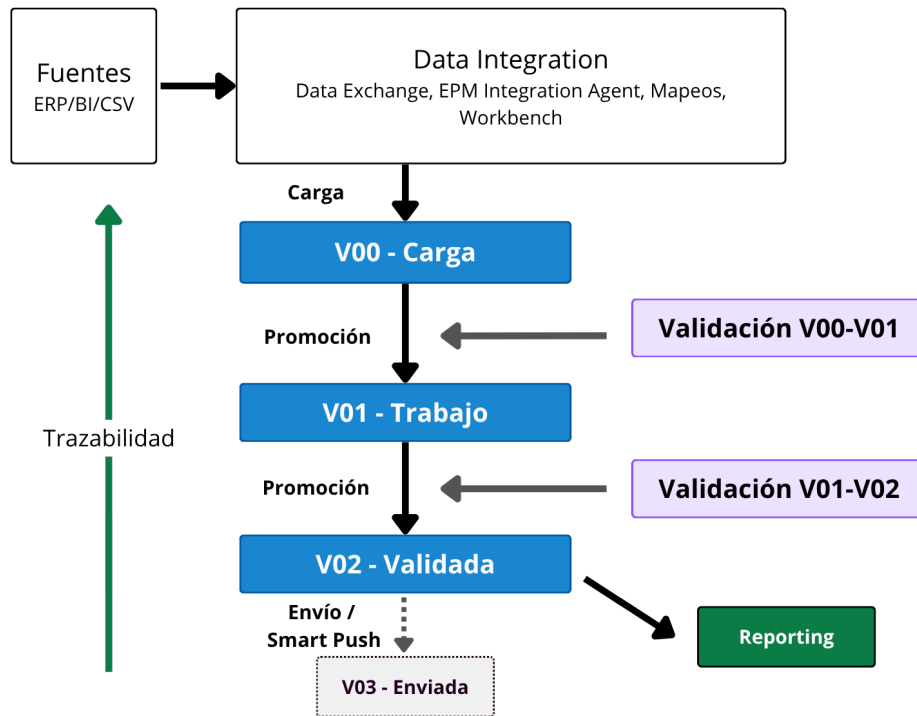
La ejecución sigue una cadena cerrada: extraer origen, aplicar mapeos, revisar Workbench, resolver incidencias y cargar en V00. Solo después se ejecutan reglas de cierre, promoción de versión y matrices de control; el Workbench valida la extracción, no sustituye al control interno del cubo.

## 6.4 Ciclo de vida del dato: versiones V00, V01 y V02

La figura 6.1 resume el tránsito del dato financiero desde las fuentes corporativas hasta su publicación. La carga inicial se concentra en V00; la revisión y los ajustes se realizan en V01; y V02 actúa como versión validada para reporting. Cuando aplica, V03 representa un estado técnico de envío o sincronización posterior.

El ciclo V00 → V01 → V02 es la capa de gobierno del dato financiero (Oracle Corporation 2026j). La matriz de validación se abre siempre **después** de promocionar, como primer paso del tramo destino; nunca anticipa la copia. La tabla 6.3 fija permisos, propósito y transición.

Una nueva extracción del ERP reinicia el ciclo en V00; V02 nunca se corrige directamente.



**Figura 6.1:** Ciclo de vida del dato financiero desde la carga inicial hasta la validación y consumo en reporting.

Fuente: Elaboración propia.

	V00 – Carga	V01 – Trabajo	V02 – Validada
<b>Escritura</b>	Integraciones, reglas, entrada manual autorizada	Usuarios autorizados para ajustes puntuales por entidad	Sin escritura; solo lectura publicada
<b>Propósito</b>	Dato bruto enriquecido: KPIs, <i>Input Balance</i> , <i>Rolling Forecast</i>	Revisión y corrección acotada; la entrada principal sigue en V00	Dato oficial para reporting
<b>Visibilidad</b>	Administradores y calidad	Roles con permisos de lectura/escritura sobre entidad	Todos los roles autorizados, incluido <i>Viewer</i>
<b>Transición</b>	Reglas en V00; DATACOPY a V01	Matriz V00–V01; ajustes; promoción por formulario a V02	Matriz V01–V02; corrección solo reabriendo V01

**Tabla 6.3:** Propiedades del protocolo de versiones V00 / V01 / V02 (Oracle Corporation 2026j).

## 6.5 Matrices de validación entre versiones

Las matrices comparan versiones dentro del cubo, complementando Workbench y *audit trail*. Se implementan como miembros de control en *EstadosFinancieros*; *Versión* va en columnas y las magnitudes de control en filas. El formulario de workflow se enlaza desde la lista de tareas del capítulo 7.

- **Secuencia:** promocionar, abrir matriz del tramo destino, revisar diferencias y habilitar ajustes o cierre.
- **Regla de diferencia:** sin ajustes en V01, la desviación debe ser cero; con ajustes, solo en intersecciones autorizadas, como Ajuste en *CriterioAjuste*.
- **Filas:** totales de P&G, balance y flujos; miembros VALID\_/VALID\_AUX; cuadro SOC\_ frente a PGC\_/IFRS\_; cuadro P&G–balance; *Input Balance*, ICP y ratios como empleados, ROA, ROI y ROE.
- **Bloqueo:** umbrales de diferencia o variación pueden activar Groovy para impedir la siguiente promoción (capítulo 8).

## 6.6 Gestión de errores y controles de calidad

---

La calidad se controla antes de la carga, dentro del cubo y después de cada escritura (Oracle Corporation 2026j).

### 6.6.1 Controles preventivos

- **Mapeos:** el Workbench retiene registros sin mapeo; si superan el 0,5 por ciento del total, se asume cambio no comunicado en plan de cuentas u origen.
- **Variación:** desviaciones superiores a  $\pm 30$  por ciento frente al mismo período del año anterior generan alerta de extracción o negocio.

### 6.6.2 Controles detectivos

- **Audit trail:** escritura automática o manual registrada con usuario, hora, valor anterior y valor nuevo.
- **Conciliación:** informes EPM frente a saldos ERP por redondeo y moneda.
- **Fallo de carga:** si el error ocurre antes de transformar, se relanza la regla completa; si los datos ya están en Workbench, se relanza solo la carga al cubo.

## Capítulo 7

# Proceso Operativo de Planificación

Este capítulo traduce la arquitectura en operación diaria: formularios para consultar o introducir datos, listas de tareas para ordenar el flujo y variables de sustitución para reutilizar los mismos artefactos en distintos años, meses, escenarios y filiales. El control técnico del dato queda definido en el capítulo 6; aquí se especifica cómo lo ejecuta el usuario.

## 7.1 Formularios

El **formulario** es la rejilla Essbase de consulta/entrada (Oracle Corporation 2025a). En esta arquitectura: POV = entidad, escenario, versión y año; *Page* = dimensiones navegables como *Línea de Negocio*, *Canal* o *Producto*; filas/columnas = *EstadosFinancieros* y *Mes*. La vista muestra alias funcionales, no códigos técnicos, según la política del apartado 4.4.3.

### 7.1.1 Tipos de formulario en la arquitectura

Familia	Uso	Regla de diseño
Entrada y revisión	Real, métricas, KPIs, <i>Budget</i> , FCST01–FCST12, proyección L/P y ajustes en V01	Editables solo en intersecciones autorizadas; V00 concentra entrada/carga y V01 ajustes acotados
Consulta agregada	Seguimiento consolidado y soporte a dashboards del apartado 11.1.1	Solo lectura; orientado a análisis, no a workflow
Workflow	Matrices de validación entre versiones y seguimiento de cierre	Siempre después de promocionar, como primer paso del tramo destino definido en la sección 6.5

Tabla 7.1: Familias de formularios en la arquitectura operativa.

### 7.1.2 Lógica asociada al formulario

El POV y las dimensiones *Page* referencian variables de sustitución (&AnoActual, &EscenarioFCST, etc.; apartado 7.3.1) para reutilizar formularios entre ciclos. Las

reglas habituales se ejecutan en **Run After Save**; las acciones puntuales, como propagación de crecimiento o Smart Push acotado, se lanzan desde menús de acción (Oracle Corporation 2026j). Se evita pedir RTP al abrir el formulario y se reserva Groovy con grid para guardado o menú, desde donde puede rellenar RTP con el PDV activo (apartado 8.3.3).

## 7.2 Listas de tareas y orquestación del proceso

---

### 7.2.1 Listas de tareas (*Task Lists*)

Las **Task Lists** ordenan el cierre por entidad y período (Oracle Corporation 2025i): indican pasos, responsables, fechas fin y estado. No sustituyen a Calc Script ni a Groovy; exponen al usuario una secuencia controlada de formularios, URL e instrucciones alineada con el ciclo  $V00 \rightarrow V01 \rightarrow V02$ .

### 7.2.2 Tipos de tarea disponibles

Planning permite tareas descriptivas, formularios, reglas de negocio, URL y aprobaciones (Oracle Corporation 2025i). En este diseño se usan tres:

- **Descriptiva:** nodo padre por tramo o bloque operativo.
- **Formulario:** consulta, entrada, matriz, ajuste y promoción por entidad.
- **URL:** acceso a Data Exchange para cargas ERP; es la única excepción al formulario como paso de trabajo.

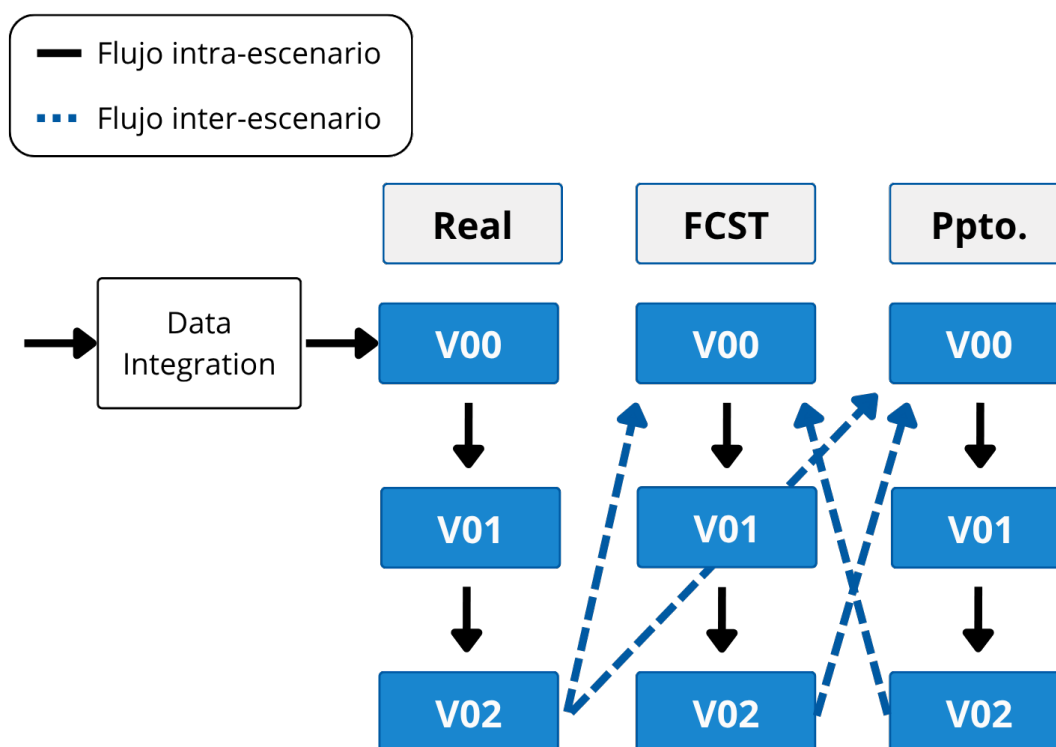
No se modelan reglas de negocio ni aprobaciones nativas como pasos de lista: la lógica batch se ejecuta desde formularios, menús de acción o planificación técnica externa (capítulo 8).

## 7.3 Flujos de ejecución por tipo de dato

---

Los flujos derivan de la dimensión *Escenario* (apartado 4.2.2) y comparten parametrización, composición y asignación por entidad. La tabla 7.4 resume las diferencias operativas.

La figura 7.1 representa la doble lógica de movimiento del dato: cada escenario mantiene un ciclo intraescenario  $V00 \rightarrow V01 \rightarrow V02$ ; además, los flujos interesenario reutilizan versiones validadas como base de ciclos posteriores de *Forecast* o presupuesto, manteniendo trazabilidad entre escenarios.



**Figura 7.1:** Flujos intraescenario e interescenario entre versiones de trabajo, validación y planificación.

Fuente: Elaboración propia.

### 7.3.1 Variables de sustitución y parametrización de flujos

Las **variables de sustitución** fijan el contexto activo de *Year*, *Mes*, *Escenario* y horizonte plurianual (Oracle Corporation 2026j). Formularios, listas y Calc Scripts referencian &NombreVariable; así se evita codificar FY25, FCST07 o PRTR26\_28 en cada artefacto.

Variable	Contenido y uso
AnoActual	Miembro activo de <i>Year</i> (FY25, FY26, ...).
MesActual	Mes o subconjunto YTD abierto (M01–M12).
EscenarioFCST	Forecast mensual activo (FCST01–FCST12; apartado 4.2.2).
EscenarioPR	Escenario de proyección L/P (PRTR24_26, PRTR25_27, ...).
Ano1Presupuesto– Ano3Presupuesto	Tres ejercicios del horizonte plurianual; se usan en POV, FIX y reglas con BegBalance.

**Tabla 7.2:** Variables de sustitución mínimas para parametrizar los flujos operativos.

Si dos filiales o calendarios ejecutan ciclos distintos, el catálogo se replica por espacio: AnoActual\_Desc1, MesActual\_Desc1, EscenarioFCST\_Desc1, etc. Los valores se mantienen desde una lista de administración del ciclo; el planificador de

filial no modifica variables.

### 7.3.2 Composición de las listas de tareas

La lista usa dos niveles: tareas padre descriptivas por tramo (V00, V01, V02) y subtareas operativas. La promoción por versión no aparece como regla aislada: se ejecuta desde un formulario con una fila por EXXXX; el menú de acción promueve solo esa entidad y registra fecha/hora.

Tipo de tarea	Uso en los flujos
Descriptiva	Agrupar el tramo de versión o bloque del proceso; no ejecuta acción.
Formulario	Entrada, revisión, matriz de validación, ajustes V01 y promoción a V02 por entidad.
URL	Lanzamiento de Data Exchange para integración ERP en Real.

**Tabla 7.3:** Tipos de tarea y su papel en las listas de tareas de la arquitectura.

	Real	Forecast	Proyección L/P
<b>Escenario</b>	Real (apdo. 4.2.2)	FCST01–FCST12 (apdo. 4.2.2)	PRTR26_28, etc. (apdo. 4.2.2)
<b>Origen</b>	ERP	Real cerrado + estimación	Hipótesis estratégicas
<b>Acción</b>	Cargar, revisar y conciliar; sin entrada manual	Completar meses abiertos en V00; ajustar opcionalmente en V01	Definir cifras pluri- anuales
<b>Tiempo</b>	Mensual	Mensual	Año 1 mensual; resto anual en BegBalance

**Tabla 7.4:** Flujos de ejecución diferenciados por tipo de dato.

### 7.3.3 Flujo del Real: carga y conciliación

Real entra en V00 por Data Exchange (capítulo 6). La lista abre con URL de integración, sigue con formularios de revisión, promociona a V01, muestra la matriz V00–V01 como primera tarea del tramo, permite ajustes de conciliación y cierra con promoción a V02.

### 7.3.4 Flujo del Forecast: avance mensual

El *Rolling Forecast* usa formularios de entrada en V00 para meses abiertos; la proyección automática y la copia a V01 no son pasos de lista. En V01, matriz, ajustes y promoción a V02; en V02, matriz final y lectura oficial, según la tabla 6.3.

### 7.3.5 Flujo de la Proyección a largo plazo

La proyección plurianual es anual o semestral: año 1 mensual y años siguientes en BegBalance (apartado 4.2.2). La lista contiene hipótesis en V00, matriz, ajustes y promoción; normalmente no requiere URL de integración.

### 7.3.6 Instanciación por entidad: holding y filiales

La dimensión *Entidad* sigue el patrón Holding → Filial (capítulo 4). Cada EXXX recibe su lista del ciclo, limitada por seguridad (capítulo 10); el holding fija calendario, fechas límite y seguimiento (Oracle Corporation 2025i).

- **Filial:** ejecuta tareas propias, revisa fechas, lanza URL cuando aplica y promueve su fila de entidad a V02.
- **Holding:** monitoriza pendiente/completado por filial y el *timestamp* de cierre o promoción.
- **Plantilla común:** una definición maestra por proceso (Real, Forecast, L/P), instanciada por filial, período y variables del ciclo.

## Capítulo 8

# Lógica de Negocio Avanzada y Orquestación

---

La lógica de negocio convierte el cubo en un sistema operativo de planificación: mueve versiones, limpia destinos, recalcula KPIs, valida formularios y ejecuta envíos entre cubos. La arquitectura separa responsabilidades para evitar reglas monolíticas:

- **Calc Script:** cálculo masivo dentro de Essbase: copias, borrados, KPIs, prorrateos y lógica post-carga.
- **Groovy:** orquestación contextual: lee formulario, usuario, PDV y celdas; decide qué lanzar y con qué RTPs.
- **Rule Set:** secuencia fija de reglas Calc, sin ramas, para procesos repetitivos.

El capítulo describe la plantilla funcional; la sintaxis ejecutable está en el anexo A. El criterio de elección Calc / Groovy / Rule Set se resume en el apartado 8.4.

## 8.1 Calc Scripts: lógica de cálculo masivo

---

Los Calc Scripts ejecutan transformaciones sobre el modelo multidimensional: no leen el formulario ni toman decisiones de contexto; operan sobre el alcance que se les pasa por variables de sustitución o RTPs. Son adecuados cuando el comportamiento es estable y el volumen de celdas es alto.

### 8.1.1 Cómo estructurar los Calc Scripts

La arquitectura usa pocas reglas genéricas, parametrizadas por ciclo o formulario (capítulos 7.3.1 y 8.3.3). Toda regla de copia o cálculo sigue tres pasos:

1. **Acotar alcance:** fijar año, mes, escenario, entidad, versión, cuentas y dimensiones analíticas mediante FIX (apartado 8.1.3).
2. **Limpiar destino:** ejecutar CLEARBLOCK o lógica equivalente para evitar mezcla con datos previos.
3. **Calcular o copiar:** aplicar fórmula, prorrateo o DATACOPY en el mismo alcance.

El patrón se reutiliza para promoción de versiones, *Rolling Forecast*, post-carga y KPIs; solo cambian las dimensiones fijadas y las cuentas afectadas.

### 8.1.2 Cabecera, dependencias y *changelog*

Cada regla incluye cabecera documental: objetivo, dimensiones afectadas, RTPs, variables, dependencias, responsable y cambios. La plantilla está en la sección A.1; los listados del anexo muestran solo el cuerpo ejecutable.

### 8.1.3 Qué trozo del cubo toca cada regla (el FIX)

El FIX define el alcance de ejecución en Essbase (Oracle Corporation 2025d). El contexto puede venir de variables del ciclo, mantenidas por el administrador, o de RTPs rellenados desde el formulario. La tabla 8.1 fija el criterio dimensional de la arquitectura.

Dimensión	Alcance habitual en reglas Calc
<i>Year</i>	Ejercicio activo: variable global o año del formulario.
<i>Mes</i>	Mes de cierre, YTD abierto o período visible.
<i>Escenario</i>	<i>Real</i> , <i>Budget</i> , FCSTnn o proyección L/P.
<i>Entidad</i>	Filial concreta desde formulario; conjunto de filiales u holding en batch.
<i>Versión</i>	Origen/destino: V00, V01, V02.
<i>Línea de Negocio,</i> <i>Producto, Canal,</i> <i>Contrapartida</i>	Normalmente totalizadores; en interco, CTP_TOT.
<i>CriterioAjuste</i>	Carga para integraciones; Ajuste para correcciones de cierre (apartado 4.2.8).
<i>EstadosFinancieros</i>	Cuentas PGC/IFRS/gestión o KPIs concretos; nunca todo el outline salvo necesidad explícita.

**Tabla 8.1:** Alcance dimensional habitual de las reglas Calc (filtro FIX).

La regla debe apuntar a nodos totales cuando se espera crecimiento orgánico del outline; así una nueva filial o cuenta queda cubierta sin reprogramar el script.

### 8.1.4 Ejemplo: promoción entre versiones (V00 → V01)

#### Qué problema resuelve

En el ciclo del capítulo 6, V00 contiene carga ERP, reglas automáticas y entrada autorizada. La promoción a V01 duplica ese estado para permitir ajustes opcionales sin alterar el dato de origen.

#### Idea general en tres pasos

La regla aplica el patrón del apartado 8.1.1: mismo alcance en origen y destino, limpieza de V01 y DATACOPY desde V00. La sintaxis está en el listado A.2. El KPI del listado A.3 usa la misma disciplina: limpiar miembro destino y recalcular.

### Quién rellena año, entidad y mes

En ejecución programada, el administrador fija año, mes y escenario mediante variables del ciclo. Desde formulario, Groovy lee filial, año, mes y escenario visibles, y los pasa como RTPs a la misma regla genérica (apartado 8.3.3).

### Qué no cubre este ejemplo

- La promoción a V02 es un cierre por filial lanzado desde formulario, no el mismo DATACOPY V00 → V01.
- La matriz V00–V01 se abre después de la copia como primera tarea del tramo V01 (sección 6.5).
- La lista de tareas no necesita exponer este Calc Script como subtarea de regla: puede ejecutarse programado o desde menú de acción.

## 8.2 Paso de contexto entre formulario, Groovy y Calc Script

El formulario no representa el cubo completo, sino una vista acotada por miembros de sus dimensiones: año, mes, versión, entidad, escenario, cuentas, productos u otras dimensiones analíticas. Esa vista fija el contexto operativo que ve el usuario y delimita qué intersecciones son relevantes para una acción concreta. La figura 8.1 muestra cómo ese contexto se convierte en parámetros de ejecución.

Los Calc Scripts no leen el *grid* del formulario ni conocen por sí mismos qué está visualizando el usuario. Su ámbito de actuación se define en Essbase mediante FIX, variables de sustitución y RTPs. Por tanto, un Calc Script es potente para ejecutar operaciones masivas, pero necesita que otro componente le entregue el alcance correcto: entidad, año, mes, escenario, versión y miembros analíticos afectados.

Groovy sí opera en la capa de formulario. Puede acceder al PDV, al *grid*, a las celdas editadas y al usuario que ejecuta la acción; con ese contexto valida datos, decide si procede lanzar una regla y construye los RTPs que recibirá Calculation Manager. La arquitectura usa los RTPs como contrato entre la capa interactiva y la capa de cálculo: el formulario aporta contexto, Groovy lo interpreta y el Calc Script lo consume dentro de un FIX controlado. Así se mantienen reglas genéricas y reutilizables sin sacrificar precisión en la ejecución.

## 8.3 Groovy Scripts: orquestación dinámica en tiempo real

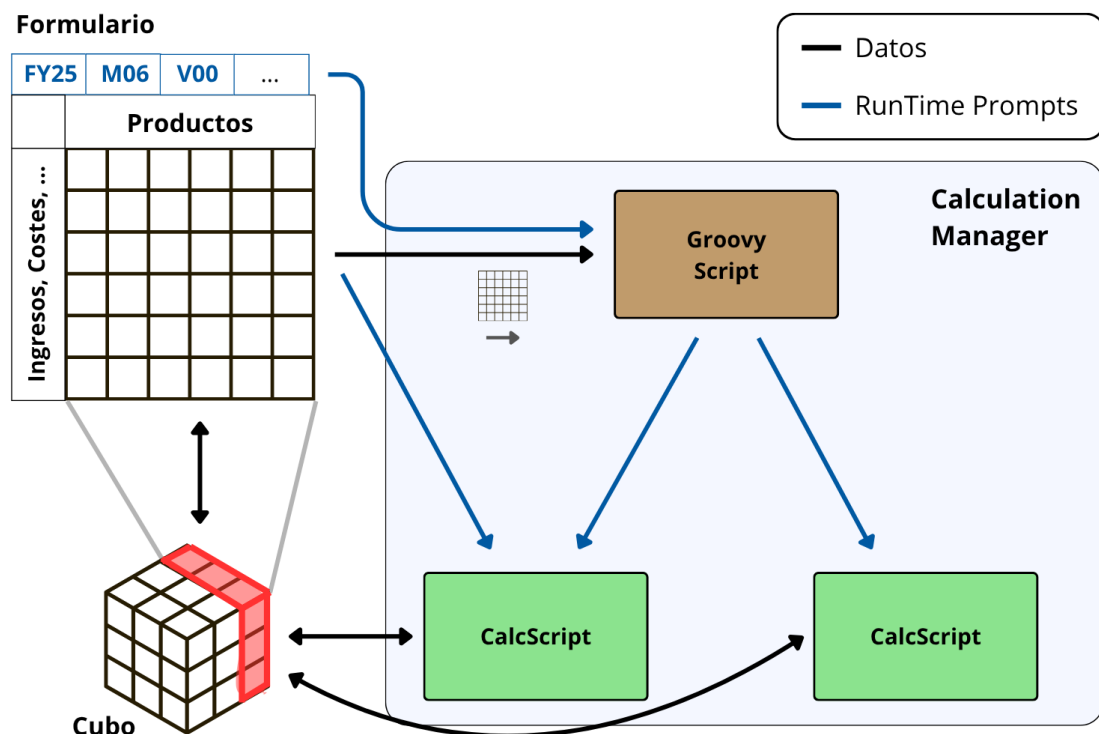


Figura 8.1: Orquestación de contexto entre formulario, Groovy, RTPs y Calc Scripts.

Fuente: Elaboración propia.

Groovy actúa en el formulario: valida guardados, decide reglas, rellena RTPs, invoca jobs REST y ejecuta *Data Maps*. El cálculo pesado permanece en Calc Script; Groovy aporta contexto y control.

### 8.3.1 Validación al guardar el formulario

En **Run After Save**, Groovy revisa celdas editadas y bloquea errores evidentes, como importes negativos no permitidos. Complementa las validaciones del formulario y evita que el dato incorrecto quede grabado. Listado de referencia: A.4.

### 8.3.2 Lanzar reglas Calc desde el formulario

Desde menú de acción o guardado, Groovy identifica escenario, entidad, año, mes y celda modificada; selecciona la regla Calc adecuada y la encola por API REST con los RTPs. Puede lanzar una regla única o iterar por filas/celdas válidas. Listados: A.5 y A.6.

### 8.3.3 Una misma regla para varios años y escenarios

La arquitectura evita duplicar reglas como `Copia_FY25_Real` o `Copia_FY26_Real`. En batch, el contexto viene de variables globales; en formulario, Groovy lo obtiene del PDV y lo pasa como RTP. Así la promoción  $V00 \rightarrow V01$  del apartado 8.1.4 sirve para cualquier año, filial y escenario.

### 8.3.4 Envío de datos entre cubos o entre filiales

Los *Data Maps* se definen una vez por administración: origen, destino y emparejamiento dimensional. Groovy los ejecuta acotados al contexto del formulario para no enviar todo el cubo: filial, año, mes, escenario y bloque funcional. Casos de referencia: Smart Push dentro de la aplicación (listado A.7) y envío filial  $\rightarrow$  grupo con `createSmartPush` (listado A.8).

## 8.4 Cuándo usar Calc Script, Groovy o Rule Set

La decisión tecnológica depende del tipo de variabilidad: datos masivos, contexto de usuario o secuencia fija. La tabla 8.2 resume el criterio.

Pregunta clave	Calc Script	Groovy	Rule Set
Qué hace	Transforma datos en Essbase	Orquesta contexto y reglas	Encadena reglas Calc
Condiciones o ramas	No; comportamiento fijo	Sí; lógica Java/Groovy	No
Parámetros	RTP o variables; no lee grid	PDV, celdas y RTP dinámicos	No
Invoca otros Calc Scripts	No	Sí; encola jobs Planning	Sí; lista fija
Uso típico	$V00 \rightarrow V01$ , KPIs, prorrateos	Validación, menú <i>Promover</i> , Smart Push	Post-carga nocturna (Oracle Corporation 2026j)

**Tabla 8.2:** Cuándo usar Calc Script, Groovy o Rule Set en la arquitectura.

Regla práctica: **Calc Script para el qué, Groovy para el cuándo y para quién, Rule Set para la secuencia fija.** Si la ejecución depende del escenario, entidad, formulario o celda, corresponde a Groovy aunque termine lanzando Calc Scripts.

## Capítulo 9

# Analítica Predictiva e Inteligencia Artificial con IPM Insights

---

La arquitectura ya centraliza datos, gobierna jerarquías, automatiza integraciones y ejecuta lógica financiera. La capa predictiva no sustituye ese diseño: lo consume. En Oracle Cloud EPM Planning conviene distinguir tres piezas: **IPM Insights**, que descubre señales estadísticas y patrones financieros; **Auto Predict**, que genera predicciones masivas sobre series históricas; y **Bring Your Own ML (BYOML)**, que permite incorporar modelos externos en formato PMML. La decisión arquitectónica es común a las tres: ninguna señal predictiva se convierte en dato oficial sin revisión humana y promoción controlada a través de  $V00 \rightarrow V01 \rightarrow V02$ .

## 9.1 Arquitectura de IPM Insights

---

IPM Insights opera dentro de Planning y automatiza la fase de descubrimiento de señales sobre datos multidimensionales: tendencias, anomalías, variaciones, sesgos de forecast y diferencias entre previsiones de usuarios y valores predichos (Oracle Corporation 2026d). Los administradores definen qué análisis ejecutar, sobre qué *slices* del cubo y con qué umbrales; los planificadores solo visualizan los insights correspondientes a los datos a los que tienen acceso (Oracle Corporation 2026g). Por tanto, IPM no se plantea como un repositorio paralelo de datos, sino como una capa de análisis sobre el mismo cubo gobernado.

En esta arquitectura, IPM Insights se usa como **capa de señal**; Auto Predict se reserva para generar predicciones y, cuando proceda, escribir resultados en una combinación de escenario y versión; BYOML se reserva para modelos externos con variables o algoritmos no cubiertos por la predicción nativa. La versión  $V00$  actúa como zona de aterrizaje de propuestas predictivas: el analista revisa, ajusta o descarta antes de promover a  $V01$  y  $V02$ .

### 9.1.1 Configuración inicial del motor

**Dimensión de tiempo:** *Period*, eje cronológico para construir y comparar series temporales.

Componente	Función principal	Uso en la arquitectura
IPM Insights	Detecta tendencias, anomalías, variaciones, sesgo y riesgo sobre datos históricos o predichos.	Priorizar revisión financiera; no escribe por sí solo el dato oficial.
Auto Predict	Genera predicciones masivas a partir de histórico y puede escribir resultados en escenario/versión.	Precargar 100% de Forecast con una propuesta estadística revisable.
BYOML / PMML	Importa modelos entrenados fuera de EPM y los despliega mediante configuración y reglas Groovy.	Incorporar drivers externos o modelos de Ciencia de Datos bajo control de Planning.

**Tabla 9.1:** Papel de IPM Insights, Auto Predict y BYOML en la arquitectura predictiva.

**Slice de análisis:** intersecciones de entidad, cuenta, escenario, versión y dimensiones analíticas sobre las que se desea generar insights.

**Ventana histórica:** 36 meses de datos reales validados; tres ejercicios completos permiten observar tres repeticiones de cada mes.

**Granularidad:** nivel hoja de *Entity* y *Account*; evita que los totales consolidados oculten comportamientos de filial o cuenta.

**Umbral de alerta:** criterio de gobierno aplicado sobre medidas de variación, sesgo, riesgo u outlier. En esta arquitectura se usa una referencia estadística equivalente a 2,5 desviaciones típicas para anomalías.

### 9.1.2 Justificación metodológica de los parámetros predictivos

Los parámetros no son valores decorativos de herramienta, sino controles de arquitectura: deben reducir incertidumbre sin generar más revisión manual que la que eliminan. El objetivo no es automatizar la aceptación del forecast, sino priorizar dónde debe mirar el analista.

**Ventana histórica de 36 meses.** Las series financieras mensuales suelen tener estacionalidad anual. Doce meses solo muestran una vuelta del ciclo; veinticuatro meses siguen siendo frágiles ante un ejercicio atípico. Una ventana de 36 meses aporta tres observaciones por posición estacional, mejora la separación entre tendencia, estacionalidad y ruido, y conserva relevancia financiera: captura cierres, campañas y calendario fiscal sin arrastrar en exceso estructuras obsoletas.

**Umbral de anomalías de 2,5 desviaciones típicas.** Oracle permite definir umbrales para activar insights; en anomalías, la señal se basa en una medida de outlier que cruza un umbral definido (Oracle Corporation 2026d). Como criterio de arquitectura, se adopta una equivalencia estadística de 2,5 desviaciones típicas sobre residuos o desviaciones normalizadas. Bajo una normal estándar:

$$P(|Z| > 2,5) = 2 \cdot (1 - \Phi(2,5)) \approx 0,0124$$

Por tanto, solo alrededor de **1,24 por ciento** de observaciones deberían marcarse como extremas sin cambio real del proceso. Dos sigma elevaría las alertas a unos 4,55 por ciento y generaría fatiga por falsos positivos en cubos con miles de intersecciones; tres sigma bajaría a 0,27 por ciento, pero perdería desviaciones materialmente relevantes. El umbral **2,5** mantiene rareza suficiente para exigir revisión y sensibilidad suficiente para no detectar solo errores extremos.

**Criterio económico de calidad predictiva.** El coeficiente de determinación mide la proporción de variabilidad explicada por el modelo (James et al. 2021). En ingresos, un  $R^2 < 0,6$  invalida la aceptación automática porque más del 40 por ciento de la variación histórica queda fuera del modelo. Esta regla se define como **política de gobierno**, no como dependencia exclusiva de una métrica nativa concreta: si el motor disponible no expone  $R^2$ , se aplica una métrica equivalente de backtesting, como MAPE, RMSE o desviación media frente al histórico. La lógica financiera es la misma: una previsión débil no debe promoverse sin explicación de drivers, porque el error de ingresos se propaga a margen bruto, compras, inventario, caja, covenants y bonus.

## 9.2 Caso de uso 1: Detección de sesgos en el proceso presupuestario

### 9.2.1 Descripción del problema

El **sesgo de planificación** aparece cuando los gestores presupuestan de forma sistemáticamente más optimista o pesimista que la realidad: sobreestiman ingresos, subestiman costes o introducen colchones para facilitar el cumplimiento posterior. El resultado es una desviación recurrente que erosiona la credibilidad del presupuesto.

## 9.2.2 Implementación con IPM Insights

IPM Insights contempla insights de variación y sesgo de forecast, comparando escenarios históricos como Forecast y Actual y activando una señal cuando la variación, el sesgo o el riesgo cruzan el umbral configurado (Oracle Corporation 2026d). En esta arquitectura, ese resultado se complementa con una métrica de gobierno calculable por cuenta y entidad:

$$\text{Sesgo} = \frac{1}{n} \sum_{t=1}^n \frac{\text{Forecast}_t - \text{Real}_t}{\text{Real}_t} \times 100$$

La serie se clasifica como optimista si el sesgo supera +5 por ciento, neutra si queda entre -5 y +5 por ciento, y pesimista si cae por debajo de -5 por ciento. La finalidad no es bloquear automáticamente la planificación, sino priorizar entidades y cuentas con desviación persistente.

## 9.2.3 Uso operativo del análisis de sesgo

El Director Financiero usa el resultado para exigir explicación en entidades con sesgo recurrente, introducir factores de corrección, ajustar hipótesis de forecast y documentar el patrón en los *Report Packages* de Narrative Reporting.

## 9.3 Caso de uso 2: Previsión automática de ingresos por series temporales

### 9.3.1 Descripción del problema

El *Rolling Forecast* exige proyectar meses abiertos a partir de reales disponibles. Si se hace manualmente, consume tiempo del equipo de Control de Gestión y hereda los sesgos del proceso presupuestario. La arquitectura usa predicción automática solo como punto de partida: la propuesta estadística entra en V00, no en la versión oficial.

### 9.3.2 Implementación con Auto Predict e IPM Insights

Auto Predict permite definir predicciones basadas en histórico, ejecutarlas de forma programada y escribir resultados en una combinación de escenario y versión (Oracle Corporation 2026c). Sus métodos incluyen opciones estacionales, no estacionales y ARIMA; la herramienta selecciona el método aplicable según la definición y el comportamiento de la serie (Oracle Corporation 2026e). Para cada combinación relevante de entidad-cuenta, la arquitectura espera:

- **Previsión central:** valor esperado por mes del horizonte abierto.
- **Escenarios de sensibilidad:** cuando estén disponibles, mejor y peor caso como referencia para análisis de riesgo.
- **Calidad de ajuste o error retrospectivo:**  $R^2$ , MAPE, RMSE o métrica equivalente usada como criterio de aceptación.

IPM Insights se usa después para comparar la previsión del planificador con valores predichos y detectar desviaciones, sesgo o riesgo (Oracle Corporation 2026d). Así se separan dos responsabilidades: Auto Predict genera una propuesta cuantitativa; IPM Insights ayuda a decidir dónde esa propuesta o el forecast humano merecen revisión.

### 9.3.3 Escritura de predicciones en el cubo

La escritura automática, cuando se active, se limita a V00 del escenario Forecast correspondiente. El ciclo estándar copia a V01, permite ajustes selectivos y promueve a V02 solo tras aprobación. Esta restricción evita que una predicción estadística sobrescriba el dato oficial y mantiene trazabilidad entre origen automático, revisión humana y versión validada.

## 9.4 Caso de uso 3: Detección de anomalías en la carga de datos

### 9.4.1 Descripción del problema

Algunos errores de integración llegan a miembros válidos del cubo, pero con importes anómalos: signo invertido, duplicidad, salto no explicado o mapeo funcionalmente incorrecto. El Workbench puede validar que la intersección existe, pero no siempre detecta que el importe carece de sentido financiero.

### 9.4.2 Implementación

IPM Insights permite detectar patrones inusuales en datos históricos que se alejan de lo esperado, por ejemplo volatilidad repentina, valores atípicos o datos reales faltantes (Oracle Corporation 2026d). En esta arquitectura, la alerta se interpreta con el umbral de 2,5 desviaciones típicas descrito en el apartado 9.1.2. El insight no corrige el dato: genera una tarea de revisión para el responsable de calidad, que decide si se trata de error de carga, evento de negocio o cambio estructural.

Este control complementa los umbrales porcentuales del capítulo 6: aquellos actúan sobre variaciones agregadas o reglas de validación; IPM aporta una señal

por intersección histórica que puede revelar errores que pasan filtros sintácticos y de mapeo.

## 9.5 Integración de modelos externos mediante BYOML

Bring Your Own ML permite importar modelos entrenados fuera de EPM en formato PMML y desplegarlos en una aplicación Planning (Oracle Corporation 2026f). El caso de uso aparece cuando Ciencia de Datos incorpora drivers externos que no residen de forma natural en el cubo: demanda macroeconómica, elasticidad de precio, riesgo de crédito, churn o señales comerciales. Tras configurar el modelo, Planning genera reglas Groovy asociadas, que pueden ejecutarse desde formularios, dashboards o jobs programados.

La condición de diseño es estricta: las entradas del modelo externo deben mapearse a dimensiones y miembros de EPM antes de permitir escritura. BYOML no elimina el gobierno del dato; lo amplía. Todo resultado externo debe aterrizar en V00, conservar trazabilidad del modelo y quedar sujeto al mismo ciclo de revisión que las predicciones nativas.

## 9.6 Limitaciones y validación humana

- **Histórico:** errores de carga, eventos no etiquetados o cambios de plan de cuentas degradan la predicción.
- **Cambio estructural:** adquisiciones, productos disruptivos, crisis de suministro o cambios de pricing invalidan patrones pasados.
- **Métrica de calidad:** la regla  $R^2 < 0,6$  debe interpretarse como umbral de gobierno; si el motor expone otra métrica, se traduce a un criterio económico equivalente.
- **Explicabilidad:** los modelos BYOML pueden ser menos transparentes que las señales nativas de IPM, por lo que requieren documentación adicional del modelo y sus drivers.

La arquitectura mantiene **validación humana obligatoria**: ninguna predicción, insight o salida BYOML se convierte en dato oficial en V02 sin revisión y aprobación explícita. La IA es soporte al juicio financiero, no sustitución del responsable del dato.

## Capítulo 10

# Modelo de Seguridad y Gobierno de Accesos

---

La seguridad de la arquitectura no se limita a impedir el acceso a la aplicación. En un modelo EPM, el riesgo principal es que un usuario pueda ver datos sensibles de otra filial, modificar una versión validada, lanzar una regla fuera de contexto o imputar valores en combinaciones dimensionales incoherentes. Por ello, el diseño se organiza en cinco capas complementarias: rol de plataforma, grupo funcional, permiso sobre artefactos, seguridad de datos por dimensión y reglas de intersección válida.

## 10.1 Roles estándar de Oracle Cloud EPM

---

Oracle Cloud EPM usa roles predefinidos para fijar el techo de capacidades de cada usuario: *Service Administrator*, *Power User*, *User* y *Viewer* (Oracle Corporation 2026b). Estos roles no sustituyen al diseño de grupos ni a la seguridad dimensional; solo determinan qué tipo de acciones puede llegar a realizar un usuario dentro del entorno.

Rol	Uso en la arquitectura
<i>Service Administrator</i>	Administración técnica: seguridad, metadatos, integraciones, reglas, refresco de cubo y auditoría. Debe limitarse a perfiles EPM/TI estrictamente necesarios.
<i>Power User</i>	Controllers corporativos: mantenimiento controlado de formularios, informes, reglas de negocio y análisis ad-hoc, siempre dentro de grupos funcionales.
<i>User</i>	Planificadores: entrada y revisión de datos mediante formularios, listas de tareas y reglas asignadas. No administra artefactos.
<i>Viewer</i>	Dirección, auditoría y usuarios de consulta: lectura de formularios, informes y dashboards asignados. No escribe ni lanza cálculo.

**Tabla 10.1:** Roles estándar de Oracle y su encaje en la arquitectura propuesta.

La asignación se gestiona desde el dominio de identidad/IAM y Access Control. La política de diseño es no asignar permisos directamente a usuarios, sino a grupos; el usuario hereda capacidades por pertenencia a grupos y por su rol predefinido.

## 10.2 Matriz de roles y permisos por artefacto

El segundo nivel controla qué artefactos puede usar cada perfil: formularios, reglas, dashboards, informes, Data Integration, Task Lists y Narrative Reporting. Oracle permite definir permisos sobre dimensiones, reglas, formularios y otros artefactos, además de intersecciones válidas y seguridad a nivel de celda (Oracle Corporation 2026a). La tabla 10.2 resume la política aplicada en este TFM.

Artefacto	Srv. Admin	Power User	User	Viewer
Metadatos y outli- ne	Mantener	Solicitar cambio	Sin acceso	Sin acceso
Data Integration	Configurar y eje- cutar	Revisar cargas	Ejecutar URL asignada si aplica	Sin acceso
Formularios	Crear y mantener	Crear/mantener corporativos	Ejecutar asigna- dos	Consultar asigna- dos
Reglas Calc/Groovy	Crear, desplegar y lanzar	Lanzar reglas corporativas	Lanzar reglas asignadas desde formulario	Sin lanzamiento
Task Lists / Task Manager	Configurar	Monitorizar	Ejecutar tareas propias	Consultar estado asignado
Dashboards e infor- mes	Publicar	Crear y consultar	Consultar asigna- dos	Consultar asigna- dos
Narrative Repor- ting	Administrar cone- xión	Mantener report packages	Aportar doclets si aplica	Consultar publica- ción final
Seguridad y audi- toría	Mantener y revi- sar	Sin mantenimien- to	Sin acceso	Lectura temporal si auditoría

**Tabla 10.2:** Matriz de permisos por rol y artefacto en la arquitectura propuesta.

## 10.3 Grupos funcionales corporativos

Los grupos traducen la organización real al cubo. En esta arquitectura, la dimensión *Entidad* sigue el patrón Holding → Filiales EXXXX; por tanto, la seguridad se asigna principalmente por filial, escenario, versión y familia de cuenta. El rol define capacidad; el grupo define alcance.

### 10.3.1 Diseño de grupos propuestos

**GRP\_ADMIN\_EPM:** Usuarios con *Service Administrator*. Mantienen aplicación, metadatos, integraciones, reglas y seguridad. No aprueban cifras de negocio.

**GRP\_DIRECCION\_FINANCIERA:** Usuarios con *Power User* o *Viewer* según función. Lectura sobre todas las entidades y escenarios; escritura solo en parámetros corporativos, ajustes del holding o comentarios ejecutivos autorizados.

**GRP\_CONTROL\_GESTION:** Usuarios con *Power User*. Lectura total del períme-

tro, ejecución de reglas corporativas, revisión de cargas y mantenimiento de formularios/informes de gestión. No modifica seguridad.

**GRP\_PLANIFICADOR\_FILIAL\_EXXXX:** Grupo específico por filial. Usuarios con *User*. Escritura en V00 y V01 de su entidad en *Budget*, FCST01–FCST12 y proyección L/P; lectura de V02 propia. Sin acceso a otras filiales.

**GRP\_CARGA\_ERP:** Usuario técnico o grupo de integración. Puede ejecutar cargas de Data Integration hacia *Real*+V00; no tiene escritura manual en formularios ni capacidad de aprobar V02.

**GRP\_COMITE\_DIRECCION:** Usuarios con *Viewer*. Lectura de informes, dashboards y Narrative Reporting únicamente sobre dato oficial V02.

**GRP\_AUDITORIA\_INTERNA:** Usuarios con *Viewer*. Lectura temporal de versiones, registros y evidencias durante ventanas de auditoría; acceso revocado al cierre de la revisión.

### 10.3.2 Principios de gestión de grupos

- **Mínimo privilegio:** cada grupo recibe solo los permisos necesarios para su tarea.
- **Sin permisos directos a usuario:** toda autorización se asigna por grupo para facilitar auditoría y rotación.
- **Separación entre capacidad y alcance:** un *Power User* no ve ni modifica todo por defecto; necesita pertenecer al grupo correspondiente.
- **Revisión periódica:** revisión semestral de usuarios, grupos y accesos temporales, coordinada con RR. HH. y Control Interno.

## 10.4 Seguridad de datos por versión, escenario y entidad

La dimensión *Version* es el eje de madurez del dato; por tanto, también es el eje de control. La arquitectura separa escritura de trabajo, ajuste y consumo ejecutivo.

El escenario *Real* tiene una restricción adicional: la carga procede de ERP hacia V00; los planificadores de filial no introducen forecast ni presupuesto sobre *Real*. En *Budget* y *Forecast*, la escritura se limita a la entidad propia, meses abiertos y cuentas planificables. Narrative Reporting consume preferentemente V02; si necesita datos preliminares, se publica como borrador interno, no como informe ejecutivo.

Versión	Propósito	Escritura permitida	Lectura
V00	Carga, propuesta automática y entrada inicial.	Integración ERP, reglas autorizadas y planificadores sobre entidad propia.	Control de Gestión, filial asignada y auditoría temporal.
V01	Revisión y ajuste controlado.	Ajustes puntuales por filial o holding según escenario.	Control de Gestión, filial asignada y auditoría temporal.
V02	Dato oficial validado.	Sin escritura de usuario ordinario; solo cierre técnico controlado.	Dirección, reporting, Narrative Reporting y usuarios autorizados.

Tabla 10.3: Política de seguridad por versión dentro del ciclo de vida del dato.

## 10.5 Intersecciones Válidas

Las **Intersecciones Válidas** no sustituyen a la seguridad; la complementan. La seguridad responde a quién puede ver o escribir. Las intersecciones válidas responden a qué combinaciones de miembros tienen sentido económico. Oracle las define como reglas que filtran intersecciones de celda en formularios, Smart View y selección de RTPs (Oracle Corporation 2026h).

### 10.5.1 Funcionamiento técnico

Cada regla define combinaciones permitidas entre dimensiones. Si una celda no cumple la regla, el usuario no debe verla como editable o el guardado debe rechazarse. El efecto práctico es doble: simplifica formularios y reduce errores de imputación antes de que lleguen a reglas Calc o matrices de validación.

### 10.5.2 Ejemplos de reglas propuestas

**Cuenta–Entidad:** cuentas operativas de tienda solo son válidas en entidades o centros de coste operativos; cuentas de holding solo en el nodo corporativo.

**Producto–Línea de Negocio:** productos de una línea solo pueden planificarse dentro de la línea correspondiente.

**Cuenta–Contrapartida:** la dimensión *Contrapartida* solo es obligatoria o válida en cuentas intercompañía; para el resto se fuerza CP\_NA.

**Escenario–Versión–CriterioAjuste:** en *Real*, V00 admite Carga; V01 admite Ajuste; V02 queda cerrada. En *Budget* y *Forecast*, V00 admite propuesta y entrada; V01, ajuste; V02, solo lectura.

**Canal–Producto:** un canal no puede planificar productos que no comercializa, evitando combinaciones vacías que inflan el espacio de entrada.

Cuando existan excepciones muy sensibles, como cuentas de retribución, bonus o reestructuración, puede aplicarse seguridad a nivel de celda. No se propone como mecanismo general porque aumenta mantenimiento; se reserva a cruces concretos con confidencialidad superior a la seguridad normal por entidad y cuenta.

## 10.6 Segregación de funciones y auditoría

La segregación de funciones evita que una misma persona pueda preparar, cargar, ajustar y aprobar un dato sin control cruzado. En esta arquitectura se formaliza mediante grupos, versiones y trazabilidad de tareas:

- el usuario técnico de carga ERP escribe en Real+V00, pero no aprueba V02;
- el planificador de filial edita V00/V01 de su entidad, pero no modifica seguridad ni metadatos;
- Control de Gestión revisa, ejecuta reglas corporativas y monitoriza cierre, pero los cambios estructurales siguen el gobierno de metadatos del capítulo 5;
- Dirección consume V02 y reportes ejecutivos, no versiones intermedias salvo autorización expresa;
- Auditoría Interna accede de forma temporal a evidencias, asignaciones y trazabilidad.

La evidencia de control se obtiene de tres fuentes: reportes de Access Control (usuarios, grupos y roles), estado de Task Lists/Task Manager y auditoría de cambios sobre datos y artefactos. El objetivo no es solo impedir accesos indebidos, sino poder demostrar quién podía hacer qué, sobre qué entidad, en qué versión y durante qué período.

## Capítulo 11

# Reporting Ejecutivo y Narrative Reporting

---

Todo el trabajo descrito en los capítulos anteriores —el modelo de datos, las integraciones, la lógica de cálculo, la IA, la seguridad— tiene un único destinatario final: la persona que tiene que tomar una decisión y necesita información fiable para hacerlo. El reporting es el punto donde la arquitectura técnica se convierte en algo visible y útil para el negocio. Si esta capa falla, o si los informes son engorrosos de actualizar o difíciles de interpretar, el valor del sistema queda enterrado bajo capas de complejidad que el usuario final no quiere gestionar.

La solución diseñada distingue dos tipos de reporting con perfiles de usuario muy distintos, y les asigna herramientas diferentes.

## 11.1 Reporting operacional en Oracle Cloud EPM Planning

Para el controller o el analista financiero que trabaja diariamente con el sistema, el propio entorno de EPM Planning es ya una herramienta de reporting suficientemente potente. No es necesario exportar datos a Excel para responder a la mayoría de las preguntas operativas del día a día; todo está disponible directamente sobre los cubos de Essbase.

### 11.1.1 Informes ad-hoc y dashboards

El controller usa el análisis ad-hoc de Planning como tabla dinámica sobre el cubo (drill por filial EXXXX, exportación a Excel) (Oracle Corporation 2026j), con la misma convención de **alias** en ejes que en los formularios (capítulo 4). En *Smart View*, las plantillas enlazadas al cubo muestran epígrafes y dimensiones con etiquetas legibles en lugar de códigos del outline. Los **dashboards** de la arquitectura combinan formularios de seguimiento, gráficos de desviación real vs presupuesto, semáforos de KPI y el estado de las listas de tareas (capítulos 7 y 10), publicados por *Power User* o administrador a cada grupo funcional.

## 11.2 Arquitectura de Oracle Narrative Reporting

---

**Narrative Reporting** elabora paquetes de informe para alta dirección (PDF o PowerPoint, con aprobación), frente al uso operativo diario de Planning (Oracle Corporation 2026j). Un **Report Package** agrupa secciones con *Doclets* de datos (enlazados a V02), de texto (comentario del CFO) y Office (plantilla con Smart View); la misma plantilla puede reutilizarse con distinta periodicidad (mensual, trimestral, anual u otra cadencia de consejo).

### 11.2.1 Conexión federada con Planning

Narrative Reporting consume Planning como *Data Source*: usuario de servicio *Viewer*, cubo de planificación y *Data Points* definidos por intersección completa (cuenta, EXXXX, escenario, versión, año, mes). Los *Doclets* de referencia se actualizan tras cada cierre validado a V02.

## 11.3 Report Package y flujo de elaboración

---

La organización puede mantener varios paquetes según el órgano de gobierno y la cadencia: informe mensual de dirección, paquete trimestral para el consejo, memorando anual de cuentas, etc. La tabla 11.1 ilustra una **plantilla tipo** de informe de resultados; los *Data Points* y el alcance temporal del paquete (mes, trimestre o ejercicio) se parametrizan al crear cada instancia del informe, sin cambiar la estructura de secciones.

Los bloques numéricos son *Reference Doclets* enlazados a V02; el análisis y la presentación ejecutiva van en *Regular* y *Office Doclets*. El informe **no arranca en Narrative Reporting**: exige antes el cierre del Real (y, si aplica, del *Forecast*) en Planning hasta V02 —integración, matrices y aprobación del capítulo 6. Solo entonces se abre el paquete correspondiente al periodo (mensual, trimestral, anual, etc.), se actualizan los *Doclets* de referencia y cada autor completa su bloque; el Director Financiero cierra el paquete antes de publicar al Comité.

## 11.4 Ventaja frente al reporting manual

---

Al anclar todos los *Doclets* numéricos a V02, el informe deja de depender de extractos Excel paralelos: un cierre validado en Planning y una actualización del paquete sustituyen días de pegado manual, con trazabilidad de autores y versión de dato en Narrative Reporting (Oracle Corporation 2026j).

Sección	Tipo de Doclet	Contenido	Autor
Resumen ejecutivo	Texto	Carta del CFO al Comité Directivo	Director Financiero
	Office (PPT)	KPIs del grupo en portada	Controller
Resultados del grupo	Referencia	Cuenta de resultados consolidada vs. Budget y vs. anterior	Controller
	Referencia	Balance de situación consolidado	Controller
	Texto	Análisis de desviaciones principales	Analista senior
Por división	Referencia	P&G por división operativa	Analista senior
	Office (PPT)	Gráficos <i>waterfall</i>	Analista senior
Perspectivas	Referencia	Forecast actualizado (resto del año)	Controller
	Texto	Hipótesis y factores de riesgo	Dir. Financiero
Anexos	Referencia	Detalle por filial y dimensiones analíticas	Analista

**Tabla 11.1:** Plantilla de secciones y *Doclets* de un Report Package de informe de resultados (periodicidad configurable).

## Capítulo 12

# Conclusiones y Líneas Futuras

---

El objetivo del TFM era diseñar una arquitectura de presupuestación y reporting corporativo capaz de reducir la dependencia de procesos manuales, reforzar la integridad del dato y aumentar la capacidad de reacción financiera. Tras el análisis comparativo del capítulo 2, la propuesta se concreta en Oracle Cloud EPM Planning y Narrative Reporting como núcleo de una arquitectura multidimensional, gobernada y extensible.

### 12.1 Cumplimiento de los objetivos

---

Los tres ejes iniciales —eficiencia operativa, integridad del dato y agilidad estratégica— están conectados: automatizar sin gobierno solo desplaza el riesgo, y gobernar sin flexibilidad reproduce modelos rígidos. La arquitectura propuesta busca resolverlos de forma conjunta.

#### 12.1.1 Eficiencia operativa

El diseño sustituye extracciones, transformaciones manuales y cálculos distribuidos en Excel por un flujo integrado en Planning. La arquitectura de integración (capítulo 6) carga el dato en V00 mediante EPM Integration Agent; la lógica de negocio (capítulo 8) concentra prorrateos, consolidación, KPIs e *Input Balance* en Calc Scripts, Groovy y reglas reutilizables. Narrative Reporting (capítulo 11) consume V02 como dato validado y reserva los *Doclets* para el análisis cualitativo. No se cuantifica ahorro real de tiempo, porque exigiría implantación productiva, pero la eliminación estructural de copias, macros y recomposición manual sí queda justificada por diseño.

#### 12.1.2 Integridad del dato

La integridad se apoya en tres mecanismos coordinados: gobierno de metadatos en Planning (capítulo 5), protocolo V00–V01–V02 (capítulo 6) y controles de seguridad/intersección válida (capítulo 10). Las jerarquías se modifican mediante validación e importación controlada; la carga, los ajustes y el dato oficial quedan separados por versión; y las combinaciones inválidas se bloquean antes de que el

usuario pueda escribir. IPM Insights (capítulo 9) añade una capa estadística para detectar anomalías en cargas reales.

### 12.1.3 Agilidad estratégica

El modelo multidimensional del capítulo 4 habilita escenarios *Budget*, *Forecast*, proyección estratégica y análisis *What-if* sin reconstruir el cubo. IPM Insights aporta previsión por series temporales para acelerar el *Forecast*; Groovy (capítulo 8) orquesta reglas desde el formulario y propaga hipótesis entre cubos o flujos afectados. La agilidad no procede de una herramienta aislada, sino de combinar escenarios, versiones, reglas parametrizadas y analítica predictiva sobre el mismo modelo gobernado.

## 12.2 Valoración crítica de la arquitectura

---

### 12.2.1 Fortalezas

- **Plataforma líder:** Oracle aporta continuidad tecnológica, soporte enterprise y posicionamiento reconocido en software FP&A.
- **Modelo escalable:** nuevas entidades, líneas, productos o canales se incorporan como miembros de jerarquía sin rediseñar formularios ni lógica base.
- **IA integrada:** IPM Insights opera sobre los datos del cubo, evitando pipelines externos de ML y latencias de sincronización.
- **Control auditable:** roles, grupos, versiones e intersecciones válidas soportan segregación de funciones y trazabilidad.

### 12.2.2 Limitaciones y riesgos

- **Vendor lock-in:** Calc Scripts, Groovy, formularios y metadatos quedan vinculados al ecosistema Oracle EPM; una migración tendría coste alto.
- **Curva técnica:** Essbase y Groovy requieren perfiles especializados, potencial cuello de botella en operación y mantenimiento.
- **Histórico para IPM Insights:** la calidad predictiva depende de disponer de datos *Actual* suficientes y consistentes (capítulo 9).
- **Seguridad compleja:** grupos e Intersecciones Válidas exigen mantenimiento disciplinado tras reorganizaciones o cambios de perímetro.

## 12.3 Líneas de trabajo futuro

El diseño base cubre planificación y reporting de gestión. Las siguientes extensiones aumentan escala, gobierno o automatización sin alterar el núcleo dimensional.

### 12.3.1 Orquestación corporativa con EPM Task Manager

El capítulo 7 usa *Task Lists*; una evolución natural sería EPM Task Manager (Oracle Corporation 2025k) cuando el grupo requiera dependencias explícitas entre tareas, plantillas de cierre reutilizables, notificaciones automáticas y seguimiento centralizado de decenas de filiales. La secuencia funcional —promoción, matriz comparativa, ajustes y cierre en V02— se mantendría; cambiaría el motor de orquestación.

### 12.3.2 Gobierno de metadatos con Oracle EDMCS

Si crece el número de aplicaciones EPM o la frecuencia de cambios estructurales, Oracle Enterprise Data Management Cloud Service (EDMCS) (Oracle Corporation 2026i) debería actuar como repositorio maestro de jerarquías. Aportaría petición, aprobación, versionado, sincronización multiaplicación y mapeo centralizado entre ERP y Planning, reutilizando la jerarquía de *Entidad* y dimensiones definidas en el capítulo 4.

### 12.3.3 Extensión a la consolidación legal (Oracle Financial Consolidation and Close)

La arquitectura cubre planificación y reporting de gestión; FCCS permitiría extenderla a consolidación legal, eliminaciones intercompañía, ajustes IFRS/PGC y estados financieros oficiales. La dimensión *Contrapartida* ya diseñada facilitaría esa transición.

### 12.3.4 Integración con arquitectura Data Mesh

En una evolución *Data Mesh*, EPM dejaría de consumir solo un ERP central y pasaría a conectarse a dominios de datos distribuidos: ventas, producción, RR. HH. o supply chain. El EPM Integration Agent permite esa conectividad; el reto sería formalizar contratos de datos entre cada dominio productor y el cubo financiero.

### 12.3.5 Modelos de lenguaje generativo (LLM) sobre los cubos de planificación

Los LLM podrían generar borradores de comentario financiero a partir de variaciones, tendencias e Insights validados. Integrados con Narrative Reporting, permitirían automatizar parte del análisis cualitativo de los *Doclets*, siempre con

revisión del Director Financiero. Esta línea conecta con la evolución generativa del ecosistema Oracle EPM y con el objetivo de cerrar el ciclo dato–decisión–informe.

## **12.4 Reflexión final**

---

La arquitectura propuesta no es solo una solución técnica: define un modelo operativo para una función financiera basada en dato único, procesos trazables, automatización gobernada e inteligencia artificial asistida. Su valor no está en sustituir el juicio financiero, sino en liberar al equipo de tareas de recolección, conciliación y reconstrucción manual para concentrarlo en análisis, decisión y comunicación ejecutiva.

# Bibliografía y fuentes consultadas

---

- Anthropic (2026). *Claude Sonnet 4.6*. Modelo de inteligencia artificial generativa empleado como apoyo auxiliar para revisión de estructura argumental, síntesis técnica y mejora de redacción. Las salidas fueron revisadas por el autor antes de su incorporación al TFM. URL: <https://www.anthropic.com/news/claude-sonnet-4-6> (visitado 04-06-2026).
- Anysphere (2026). *Composer 2.5*. Asistente de programación con inteligencia artificial empleado como apoyo auxiliar para edición técnica, revisión de estructura de código y asistencia en refactorización documental. Las salidas fueron revisadas por el autor antes de su incorporación al TFM. URL: <https://cursor.com/changelog/composer-2-5> (visitado 04-06-2026).
- Crowder, Regina y Sid Sahoo (dic. de 2025). *Magic Quadrant for Financial Planning Software*. Inf. téc. Licensed for Distribution. Revised 3 December 2025. Gartner. URL: <https://www.gartner.com/doc/reprints?id=1-2MEUQ7BC&ct=251203&st=sb>.
- Deloitte (dic. de 2025). *Technology Transformation Emerges as a Top Priority for CFOs in 2026: Q4 2025 CFO Signals Survey*. Inf. téc. Muestra: 200 directores financieros en Norteamérica con ingresos  $\geq 1.000$  M USD. Deloitte. URL: <https://www.deloitte.com/us/en/about/press-room/deloitte-q4-2025-cfo-signals-survey.html>.
- FTI Consulting (ene. de 2025). *2025 Global CFO Report*. Inf. téc. FTI Consulting. URL: <https://www.fticonsulting.com/-/media/files/insights/reports/2025/jan/global-cfo-survey-2025.pdf>.
- (mar. de 2026). *2026 Global CFO Survey Report*. Inf. téc. Encuesta a directores financieros globales. La previsión financiera (90 %) y la automatización del reporting (88 %) son las capacidades de IA prioritarias para los CFOs en 2026. FTI Consulting. URL: <https://cache.pressmailing.net/content/0cd8e2d7-bbf6-406e-9318-f748a1049bdf/global-cfo-survey-2026.pdf>.
- Google DeepMind (2026). *Gemini 3.5 Flash*. Modelo de inteligencia artificial generativa empleado como apoyo auxiliar para contraste de redacción, reformulación de pasajes técnicos y comprobación de consistencia narrativa. Las salidas fueron revisadas por el autor antes de su incorporación al TFM. URL: <https://>

- [//deepmind.google/models/model-cards/gemini-3-5-flash/](https://deepmind.google/models/model-cards/gemini-3-5-flash/) (visitado 04-06-2026).
- Hevner, Alan R. et al. (2004). «Design Science in Information Systems Research». En: *MIS Quarterly* 28.1. Ficha oficial en AIS eLibrary con enlace de descarga; se evita usar el DOI como URL principal por redirigir a una pasarela de pago, págs. 75-105. DOI: [10.2307/25148625](https://doi.org/10.2307/25148625). URL: <https://aisel.aisnet.org/misq/vol28/iss1/6/>.
- James, Gareth et al. (2021). *An Introduction to Statistical Learning: With Applications in R*. 2.<sup>a</sup> ed. Web oficial del libro con descarga gratuita del PDF de la segunda edición; referencia estadística para métricas de ajuste, regresión y evaluación de modelos predictivos. Springer. URL: <https://www.statlearning.com/>.
- KPMG (mayo de 2026). *Global AI in Finance 2026*. Inf. téc. Encuesta a 1.013 directivos financieros en 13 sectores y 20 países. KPMG International. URL: <https://assets.kpmg.com/content/dam/kpmgsites/xx/pdf/2026/05/global-ai-in-finance-report.pdf>.
- McKinsey & Company (jul. de 2024). *Toward the long term: CFO perspectives on the future of finance*. Encuesta global a directores financieros (n = 200 empresas con ingresos >1.000 M USD). URL: <https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/toward-the-long-term-cfo-perspectives-on-the-future-of-finance>.
- Mordor Intelligence (feb. de 2026). *Enterprise Performance Management (EPM) Market Size, Trends Report & Share 2026–2031*. Inf. téc. Actualizado a enero de 2026. Tamaño de mercado 2026: 7.690 M USD; previsión 2031: 11.890 M USD; CAGR 9,11 %. Mordor Intelligence. URL: <https://www.mordorintelligence.com/industry-reports/enterprise-performance-management-market>.
- OpenAI (2026). *ChatGPT 5.5*. Modelo de inteligencia artificial generativa empleado como apoyo auxiliar para validación de estructura, contraste de coherencia y asistencia en redacción académica. Las salidas fueron revisadas por el autor antes de su incorporación al TFM. URL: <https://help.openai.com/es-es/articles/11909943-gpt-55-en-chatgpt> (visitado 04-06-2026).
- Oracle Corporation (2024). *Best Practices for Designing Oracle EPM Dimensions*. Vídeo oficial Oracle EPM (YouTube). Recoge las directrices de diseño dimensional: orden *hourglass* modificado en el outline, límite de 12 dimensiones por cubo, separación de cubos BSO/ASO, combinación de dimensiones no relacionadas, jerarquías alternativas, dimensiones de atributo y Smart Lists. URL: <https://www.youtube.com/watch?v=SlPqIiWlciI>.
- (2025a). *Designing Forms — Oracle Fusion Cloud EPM Planning*. Tutorial oficial de diseño de formularios (Forms 2.0). Cubre la asociación de formularios a cubos,

el diseño del Point of View, filas y columnas, las reglas de validación y formato, y las reglas de negocio invocables desde el formulario. URL: [https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/planning\\_tutorial\\_designing\\_forms/index.html](https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/planning_tutorial_designing_forms/index.html).

- Oracle Corporation (2025b). *Managing Dimensions in Cloud EPM Business Processes*. Tutorial oficial Oracle Cloud EPM. Cubre tipos de dimensión, opciones de almacenamiento (Dynamic Calc, Store, Never Share, Label Only, Shared), dimensiones requeridas (Account, Entity, Period, Scenario, Version, Year) y dimensión Currency para aplicaciones multicorporativas. URL: <https://docs.oracle.com/en/cloud/saas/enterprise-performance-management-common/epm-tutorial-dimensions/index.html>.
- (2025c). *Oracle Cloud EPM: Design Best Practices — Hybrid BSO Design*. Parte de la guía EPM Best Practices. Detalla las mejores prácticas para la configuración del modo híbrido, incluyendo criterios de dimensiones dinámicas y densidad de bloques. Oracle Corporation. URL: [https://docs.oracle.com/en/cloud/saas/enterprise-performance-management-common/ebest/designing\\_hybrid\\_bso\\_overview.html](https://docs.oracle.com/en/cloud/saas/enterprise-performance-management-common/ebest/designing_hybrid_bso_overview.html).
- (2025d). *Oracle Essbase: Creating and Managing Essbase Applications — Optimized Hybrid Mode Cubes*. Essbase 21c. Incluye las directrices sobre clasificación de dimensiones densas y dispersas en aplicaciones de planificación financiera. Oracle Corporation. URL: <https://docs.oracle.com/en/database/other-databases/essbase/21/esscd/create-optimized-hybrid-cubes.html>.
- (2025e). *Oracle Essbase: Database Administrator's Guide — Designing Essbase Applications*. Essbase 21c. Incluye las directrices de selección de tipo de cubo BSO, ASO e Híbrido. Oracle Corporation. URL: <https://docs.oracle.com/en/database/other-databases/essbase/21/essdm/hybrid-mode-fast-analytic-processing.html>.
- (2025f). *Oracle Essbase: Database Administrator's Guide — Dimensions and Members*. Essbase 21c. Detalla los conceptos de generación (distancia desde la raíz) y nivel (distancia desde las hojas), así como la estructura jerárquica de dimensiones. Oracle Corporation. URL: <https://docs.oracle.com/en/database/other-databases/essbase/21/essdm/dimensions-and-members.html>.
- (2025g). *Oracle Essbase: Database Administrator's Guide — Options for How Members Store Data Values*. Essbase 21c. Define Store, Dynamic Calc, Label Only, Shared y Never Share; describe el *implied share* y las reglas de miembros compartidos explícitos. Oracle Corporation. URL: <https://docs.oracle.com/en/database/other-databases/essbase/21/essdm/options-how-members-store-data-values.html>.

- Oracle Corporation (2025h). *Oracle Essbase: Database Administrator's Guide — Understanding Member Consolidation Operators*. Essbase 21c. Manual PDF oficial completo, Database Administrator's Guide for Oracle Essbase, part number F17141-19, actualización del 1 de diciembre de 2025. Incluye los operadores de consolidación (+, -, \*, /, %, ~, ^) y el comportamiento de miembros compartidos. Oracle Corporation. URL: <https://docs.oracle.com/en/database/other-databases/essbase/21/essdm/database-administrators-guide-oracle-essbase.pdf>.
- (2025i). *Oracle Fusion Cloud EPM: Administering Planning — Administering Task Lists*. Documentación oficial sobre listas de tareas: guían a los usuarios a través del proceso de planificación mediante tareas, instrucciones y fechas de fin. Oracle Corporation. URL: [https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/administering\\_tasks.html](https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/administering_tasks.html).
- (2025j). *Oracle Fusion Cloud EPM: Administering Planning — Naming Restrictions for Dimensions, Members, and Aliases*. Restricciones de nomenclatura oficiales: límite de 80 caracteres, caracteres prohibidos, palabras reservadas de Calc Script y requisitos de coincidencia de miembros compartidos. Oracle Corporation. URL: [https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/restrictions\\_for\\_dimensions\\_members\\_aliases\\_and\\_forms.html](https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/restrictions_for_dimensions_members_aliases_and_forms.html).
- (2025k). *Oracle Fusion Cloud EPM: Administering Planning — Task Manager Overview*. EPM Task Manager es la opción predeterminada de flujo de tareas en aplicaciones nuevas. Soporta plantillas reutilizables, dependencias entre tareas, roles de aprobador, paneles de seguimiento y notificaciones por correo. Oracle Corporation. URL: [https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/cm\\_close\\_process\\_overview.html](https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/cm_close_process_overview.html).
- (2026a). *Oracle Fusion Cloud EPM — Setting Up Security in Cloud EPM*. Tutorial oficial sobre configuración de seguridad en Cloud EPM: roles, grupos, permisos de artefactos, reglas, formularios, intersecciones válidas y seguridad a nivel de celda. Oracle Corporation. URL: <https://docs.oracle.com/en/cloud/saas/enterprise-performance-management-common/epm-tutorial-security/index.html>.
- (2026b). *Oracle Fusion Cloud EPM — Understanding Predefined Roles*. Documentación oficial sobre los roles predefinidos de Cloud EPM: Service Administrator, Power User, User y Viewer. Oracle Corporation. URL: [https://docs.oracle.com/en/cloud/saas/enterprise-performance-management-common/cgsad/4\\_managing\\_epm\\_cloud\\_users\\_roles\\_predefined\\_roles.html](https://docs.oracle.com/en/cloud/saas/enterprise-performance-management-common/cgsad/4_managing_epm_cloud_users_roles_predefined_roles.html).

- Oracle Corporation (2026c). *Oracle Fusion Cloud EPM Planning — About Auto Predict*. Documentación oficial sobre Auto Predict: definición de predicciones, ejecución programada, predicción masiva de celdas y escritura de resultados en escenario y versión. Oracle Corporation. URL: [https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/auto\\_predict\\_about.html](https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/auto_predict_about.html).
- (2026d). *Oracle Fusion Cloud EPM Planning — About IPM Insights*. Documentación oficial sobre IPM Insights: análisis de tendencias, anomalías, variación, sesgo de previsión y revisión de insights en dashboard. Oracle Corporation. URL: [https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/insights\\_about.html](https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/insights_about.html).
- (2026e). *Oracle Fusion Cloud EPM Planning — Auto Predict Methods Options*. Documentación oficial sobre métodos de Auto Predict: métodos estacionales, no estacionales y ARIMA. Oracle Corporation. URL: [https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/auto\\_predict\\_methods.html](https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/auto_predict_methods.html).
- (2026f). *Oracle Fusion Cloud EPM Planning — Bring Your Own ML: Machine Learning Model Import*. Documentación oficial sobre importación de modelos PMML en Planning y generación de reglas Groovy asociadas al modelo. Oracle Corporation. URL: [https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/ml\\_about.html](https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/ml_about.html).
- (2026g). *Oracle Fusion Cloud EPM Planning — Configuring IPM Insights*. Documentación oficial sobre configuración de IPM Insights, definición de slices, tipos de análisis, ejecución programada y relación con Auto Predict. Oracle Corporation. URL: [https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/insights\\_configuring.html](https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/insights_configuring.html).
- (2026h). *Oracle Fusion Cloud EPM Planning — Understanding Valid Intersections*. Documentación oficial sobre intersecciones válidas en Planning y filtrado de combinaciones de celda según reglas definidas. Oracle Corporation. URL: [https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/val\\_comb.html](https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/val_comb.html).
- (2026i). *Oracle Fusion Cloud EPM: Administering and Working with Oracle Enterprise Data Management Cloud*. PDF oficial completo. Part number E93969-79. Actualización del 9 de abril de 2026. Copyright © 2018, 2026, Oracle and/or its affiliates. Oracle Corporation. URL: <https://docs.oracle.com/en/cloud/saas/enterprise-data-management-cloud/dmcaa/GUID-7807DB45-355D-4D01-88CB-2BAD75C433D4.pdf>.
- (2026j). *Oracle Fusion Cloud EPM: Administering Planning*. PDF oficial completo. Part number E94139-92. Actualización del 2 de abril de 2026. Copyright © 2001,

2026, Oracle and/or its affiliates. Oracle Corporation. URL: <https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusa/EPM-INFORMATION-DEVELOPMENT-TEAM-E94139-6691CB58.pdf>.

Peppers, Ken et al. (2007). «A Design Science Research Methodology for Information Systems Research». En: *Journal of Management Information Systems* 24.3, págs. 45-77. DOI: [10.2753/MIS0742-1222240302](https://doi.org/10.2753/MIS0742-1222240302). URL: <https://doi.org/10.2753/MIS0742-1222240302>.

## Capítulo A

# Ejemplos de código

---

Este anexo recoge el **código de referencia** de las plantillas descritas en funcional en el capítulo 8. Aquí aparece la sintaxis Essbase y Groovy; el capítulo explica **qué hace cada regla y cuándo se usa**. Autor, fechas y nombres de miembros son **placeholders genéricos**.

## A.1 Plantilla de cabecera (Calc Script y Groovy)

---

En implantación EPM, **cada regla** incluye al inicio la misma cabecera en comentarios. No forma parte de la lógica de Essbase ni de Groovy: sirve para que quien mantenga el cubo sepa qué parametriza la regla, qué dimensiones toca y qué cambios se han hecho sin abrir el historial de Planning.

- **Parámetros que pide la regla** (si los hay) o indicación de que salen del formulario.
- **Dimensiones afectadas** (año, entidad, escenario, cuentas, etc.).
- **Descripción, autor y dependencias** (otras reglas, mapas, jobs de integración).
- **Historial de cambios** (fecha, responsable, qué se modificó).

Los listados de las secciones A.2 y A.3 **omiten** esta cabecera y muestran solo el cuerpo ejecutable. En el repositorio de Planning la cabecera va siempre pegada encima del código correspondiente.

**Listado A.1:** Plantilla genérica de cabecera (comentarios al inicio de cada regla)

```

/* RTPs: {RTP_Entidad}, {RTP_Mes}, ... */
/*===== DIMENSIONES =====
   Year, Entity, Scenario, EstadosFinancieros, Mes, ...
=====
   Descripcion :
       Resumen en una o dos líneas de lo que hace la regla.
   Autor      : Responsable mantenimiento
   Dependencias: otras reglas Calc; Data Maps; &AnoActual; ...
=====*/
/*===== CHANGELOG =====
   Fecha      | Responsable | Descripcion del cambio
-----|-----|-----
   DD/MM/AAAA | Responsable | alta inicial de la regla.
   DD/MM/AAAA | Responsable | ajuste de alcance FIX / mapa.
=====*/

/* --- A partir de aqui: cuerpo de la regla (listados siguientes) --- */

```

## A.2 Calc Scripts

Plantillas de **cálculo y copia en el cubo** (cap. 8.1): promoción de versión y fórmula de KPI.

**Listado A.2:** Copiado de versión: CLEARBLOCK en destino y DATACOPY V00 → V01

```
/* 1) Vaciar version destino */
FIX(
  "Real",
  &AnoActual,
  @RELATIVE("LIN_TOT", 0),
  @RELATIVE("PRO_TOT", 0),
  @RELATIVE("CAN_TOT", 0),
  @RELATIVE("CTP_TOT", 0),
  "Carga",
  {RTP_Entidad},
  @RELATIVE("PGC", 0), @RELATIVE("Metricas", 0),
  {RTP_Mes},
  "V01"
)
  CLEARBLOCK ALL;
ENDFIX;

/* 2) Copiar origen -> destino */
FIX(
  "Real",
  &AnoActual,
  @RELATIVE("LIN_TOT", 0),
  @RELATIVE("PRO_TOT", 0),
  @RELATIVE("CAN_TOT", 0),
  @RELATIVE("CTP_TOT", 0),
  "Carga",
  {RTP_Entidad},
  @RELATIVE("PGC", 0), @RELATIVE("Metricas", 0),
  {RTP_Mes}
)
  DATACOPY "V00" TO "V01";
ENDFIX;
```

**Listado A.3:** Cálculo con fórmula: CLEARBLOCK del destino y IF/ENDIF en V00

```
/* 1) Vaciar cuenta destino */
FIX(
  &AnoActual,
  &EscenarioFCST,
  "Carga",
  {RTP_Entidad},
  "KPI_Margen_Calc",
  "BegBalance",
  "V00",
  @RELATIVE("LIN_TOT", 0),
  @RELATIVE("PRO_TOT", 0),
  "CAN_DF",
  "LIN_DF"
)
  CLEARBLOCK ALL;
ENDFIX;

/* 2) Calcular */
FIX(
  &AnoActual,
  &EscenarioFCST,
  "Carga",
  {RTP_Entidad},
  "KPI_Margen_Calc",
  "BegBalance",
  @RELATIVE("LIN_TOT", 0),
  @RELATIVE("PRO_TOT", 0),
  "CAN_DF",
  "LIN_DF"
)
  "V00" (
    IF("KPI_Indice_Periodo" != #MISSING)
      "KPI_Margen_Calc" =
        (1 + "KPI_Margen_YTD") / (1 + "KPI_Indice_Periodo") - 1;
    ELSE
      "KPI_Margen_Calc" = "KPI_Margen_YTD";
    ENDIF
  )
ENDFIX;
```

## A.3 Groovy Scripts

Plantillas de **orquestación en formulario** (cap. 8.3): validación, lanzamiento de Calc, *Data Maps*.

**Listado A.4:** Validación en guardado (*Run After Save*)

```
if (!operation.hasGrid()) {
    throwVetoException("Este script requiere un formulario con datos
guardados.")
}

DataGrid dataGrid = operation.getGrid()

dataGrid.dataCellIterator().each { DataCell cell ->

    if (cell.isMissing()) {
        return
    }
    if (!cell.editable) {
        return
    }
    if (cell.data < 0) {
        String cuenta = cell.getMemberName("EstadosFinancieros")
        cell.data = cell.existingData
        operation.warnings.add(
            "Importe negativo no permitido en ${cuenta}.")
    }
}
```

**Listado A.5:** Orquestación desde menú de acción: lanzar Calc Script vía API REST

```
if (!operation.hasGrid()) {
    throwVetoException("Este script requiere un formulario con datos
    guardados.")
}

DataGrid dataGrid = operation.getGrid()

String escenario = dataGrid.pov.find {
    it.dimensionName == "Scenariio" }?.memberName

String reglaCalc = escenario?.startsWith("FCST")
    ? "Calc_Rolling_Forecast"
    : "Calc_Promocion_V00_V01"

String rulePayload = json([
    "jobType"    : "Rules",
    "jobName"    : reglaCalc,
    "parameters": [
        "RTP_Entidad"    : rtps.RTP_Entidad.enteredValue,
        "RTP_Escenario" : rtps.RTP_Escenario.enteredValue,
        "RTP_Year"       : rtps.RTP_Year.enteredValue,
        "RTP_Periodo"    : rtps.RTP_Periodo.enteredValue
    ]
])

HttpResponse<String> response = operation.application
    .getConnection("Conexion_Planning_REST")
    .post("/HyperionPlanning/rest/v3/applications/PresupuestoPlan/jobs")
    .header("Content-Type", "application/json")
    .body(rulePayload)
    .asString()

println "Rule response: ${response.body}"
operation.messages.add("Ejecutada ${reglaCalc}.")
```

**Listado A.6:** Orquestación desde rejilla: POST REST por celda KPI válida

```
if (!operation.hasGrid()) {
    throwVetoException("Este script requiere un formulario con datos
    guardados.")
}

DataGrid dataGrid = operation.getGrid()
Dimension dimensionEF = operation.application.getDimension("
    EstadosFinancieros")

dataGrid.dataCellIterator().each { DataCell cell ->

    if (cell.isMissing()) {
        return
    }

    String cuentaEF = cell.getMemberName("EstadosFinancieros")
    if (dimensionEF.getMember(cuentaEF)?.name != "KPI_Margen_Calc") {
        return
    }
    if ((cell.data?.intValue() ?: 0) == 0) {
        return
    }

    String rulePayload = json([
        "jobType"    : "Rules",
        "jobName"    : "Calc_Ajuste_KPI_Fila",
        "parameters": [
            "RTP_Entidad": cell.getMemberName("Entity"),
            "RTP_Year"   : rtps.RTP_Year.enteredValue,
            "RTP_Mes"    : cell.getMemberName("Period")
        ]
    ])

    HttpResponse<String> response = operation.application
        .getConnection("Conexion_Planning_REST")
        .post("/HyperionPlanning/rest/v3/applications/PresupuestoPlan/jobs")
        .header("Content-Type", "application/json")
        .body(rulePayload)
        .asString()

    println "Rule response: ${response.body}"
}
```

**Listado A.7:** Smart Push acotado por entidad y período del formulario activo

```
if (!operation.hasGrid()) {
    throwVetoException("Este script requiere un formulario con datos
    guardados.")
}

DataGrid dataGrid = operation.getGrid()
def dataMap = operation.application.getDataMap("DataMap_Workforce_to_Plan")

dataMap.clearFilters()
dataMap.addFilter("Entity", dataGrid.pov
    .find { it.dimensionName == "Entity" }?.memberName)
dataMap.addFilter("Period", dataGrid.pov
    .find { it.dimensionName == "Period" }?.memberName)

dataMap.execute(false)

operation.messages.add(
    "Costes de personal actualizados correctamente en el cubo financiero.")
```

**Listado A.8:** Envío filial → grupo: una porción con createSmartPush

```
String nombreDataMap = "DM_Filial_a_grupo"

if (!operation.application.hasDataMap(nombreDataMap)) {
    throwVetoException("No existe el Data Map " + nombreDataMap)
}

/* Origen/destino del mapa definidos en admin; aqui solo el alcance del
envio. */
operation.application.getDataMap(nombreDataMap)
    .createSmartPush().execute([
        "Entidad"           : rtps.RTP_Entidad.enteredValue,
        "Ano"               : rtps.RTP_Year.enteredValue,
        "Escenario"        : rtps.RTP_Escenario.enteredValue,
        "Mes"               : rtps.RTP_Mes.enteredValue,
        "EstadosFinancieros": "ILvl0Descendants(PL)",
        "Contrapartida"     : "CTP_NA",
        "CriterioAjuste"    : "Carga"
    ], true)

operation.messages.add(
    "Enviada porcion PyG de " + rtps.RTP_Entidad.enteredValue +
    " a instancia grupo.")
```