



GRADO EN MATEMÁTICA COMPUTACIONAL

ESTANCIA EN PRÁCTICAS Y PROYECTO FINAL DE GRADO

Superficies minimales

Autor:
Eric FERRER ESCUIN

Supervisor:
José SALOM
Tutor académico:
Vicent GIMENO GARCIA

Fecha de lectura: __ de Julio de 2018
Curso académico 2017/2018

Resumen

Este documento trata sobre la estancia en prácticas en la Autoridad Portuaria de Castellón y el trabajo final de grado de la asignatura MT1030-Prácticas Externas y Proyecto Fin de Grado del grado de Matemática Computacional de la Universidad Jaume I.

El proyecto de la estancia en prácticas consiste en la creación de una aplicación de inteligencia de negocio en QlikView para analizar los datos extraídos de una base de datos Oracle.

El trabajo consiste en demostrar un teorema por el que podremos representar superficies minimales partiendo de dos funciones complejas que cumplan una serie de condiciones analíticas.

Palabras clave

Superficies minimales, funciones holomorfas, parametrización isoterma, parametrización de Weierstrass.

Keywords

Minimal surfaces, holomorphic maps, isothermic parametrization, Weierstrass parametrization.

Índice general

1. Introducción	9
1.1. Contexto y motivación del proyecto	9
2. Estancia en prácticas	11
2.1. Introducción	11
2.2. Objetivos del proyecto formativo	11
2.3. Explicación detallada del proyecto realizado en la empresa	12
2.3.1. Metodología y definición de tareas	12
2.3.2. Planificación temporal de las tareas	14
2.3.3. Estimación de recursos del proyecto	21
2.3.4. Grado de consecución de los objetivos propuestos	21
2.3.5. Conclusiones	25
3. Memoria TFG	27
3.1. Desarrollo del TFG	27
3.1.1. Funciones en variable compleja	27

3.1.2. Superficies regulares	33
3.1.3. Superficies minimales i su obtención por la parametrización de Weierstrass	39
3.1.4. Construcción de ejemplos de superficies minimales	42
4. Conclusiones	49
A. Anexo I	53
A.1. Código SQL	53
A.1.1. Vista APC_ESCALAS	53
A.1.2. Vista APC_ATRAQUES	55
A.1.3. Vista APC_MANIFIESTOS	56
A.1.4. Vista ESCALAS	61
A.1.5. Vista ATRAQUES	61
A.1.6. Vista MANIFIESTOS	62
A.1.7. Vista FACTURAS_DP y código auxiliar	63
A.1.8. Vista EXPEDIENTES	65
A.1.9. Vista ACTIVIDADES	66
A.1.10. Vista OCUPACIONES	67
A.1.11. Vista BIEN_PUBLICO	67
A.1.12. Vista CIFRAS_NEGOCIO	68
A.1.13. Vista ESTIBA	68
A.1.14. Vista DETALLE_ESTIBA	69
A.1.15. Vista APC_FACTURAS	70

A.1.16. Vista FACTURAS	72
A.1.17. Vista LINEAS_FACTURA	73
A.2. Código script QlikView	74
A.2.1. Main	74
A.2.2. Aux	77
A.2.3. Facturas	79
A.2.4. Dominio Público	81
A.2.5. Operativa Portuaria	85

Capítulo 1

Introducción

1.1. Contexto y motivación del proyecto

La idea del trabajo es ver una forma sencilla de construir ejemplos de superficies minimales partiendo de dos funciones complejas que cumplan una serie de condiciones analíticas.

Como dice J. Pérez en [6]

Las superficies minimales, junto con el problema isoperimétrico, están entre los problemas geométricos más antiguos y estudiados en Matemáticas, y su interés a través de los siglos ha provocado el desarrollo de diversas ramas de esta ciencia. En sus orígenes, el Cálculo de Variaciones desarrollado por Euler y Lagrange en el siglo XVIII permitió una formulación satisfactoria del problema de minimizar el área de una superficie con un borde dado, aunque con poca profusión de ejemplos. Un siglo más tarde, matemáticos de la talla de Enneper, Scherk, Schwarz, Riemann y Weierstrass produjeron enormes avances sobre superficies minimales mediante la aplicación del recién creado Análisis Complejo.

El objetivo del trabajo es la demostración del teorema principal (teorema 3.1.3.5). Este teorema es un teorema clásico en la teoría de superficies minimales que requiere para su demostración conocimientos en análisis complejo y también de los conceptos básicos de geometría diferencial de superficies.

Para todo ello nos basaremos en el libro de Osserman, [4], y para poder ver las demostraciones introduciremos los conceptos necesarios de variable compleja y superficies regulares necesarios.

Por la parte de variable compleja, las bases las hemos visto durante el grado en la asignatura de “Análisis complejo y de Fourier”, y posteriormente los ampliamos con conceptos nuevos de los libros de Churchill [1] y Rudin [2].

Por la parte de superficies nos basamos en el libro de Osserman ya mencionado anteriormente, además de algunos conceptos estudiados en el grado en la asignatura “Aplicaciones de la geometría diferencial y topología”, para luego también ampliarlos con nuevos conceptos del libro de Do Carmo [3].

Para acabar el trabajo, programamos la representación de las superficies y para ver los ejemplos más conocidos de superficies minimales nos basamos en el “survey” de Pérez y Meeks [5].

Finalmente, además del trabajo de superficies minimales, también se explica la estancia en prácticas en la Autoridad Portuaria de Castellón, que consistieron en crear vistas en la base de datos Oracle que ellos usan mediante uso del SQL y posteriormente la creación de una aplicación de inteligencia de negocio mediante el programa QlikView.

Capítulo 2

Estancia en prácticas

2.1. Introducción

La estancia en prácticas la he realizado en la Autoridad Portuaria de Castellón, empresa encargada de la gestión portuaria, tanto del movimiento de mercancías como de los terrenos asignados al puerto. Ellos se encargan de gestionar las tasas asociadas a carga y descarga de mercancías, así como las asociadas al uso de los terrenos del puerto. Todas estas actividades se registran en una base de datos de Oracle, que comparte la estructura básica de tablas y relaciones con el resto de puertos españoles, pues luego se extraen los datos para las estadísticas anuales del país. También se encargan del mantenimiento de las infraestructuras del puerto, así como de la expansión y desarrollo de éstas. Además se ocupan de la integración del puerto con la ciudad, promoviendo actividades, tanto deportivas como culturales, en el puerto, para así aumentar la capacidad de generar ingresos del propio puerto así como de la zona portuaria.

2.2. Objetivos del proyecto formativo

Como he comentado anteriormente, los datos del puerto se almacenan en una base de datos Oracle, con una estructura básica compartida con el resto de puertos españoles, por lo que todos deben actualizarse simultáneamente si se cambia esta estructura a nivel nacional. Recientemente, en 2015, se cambió de la estructura de la base de datos anterior, a una nueva llamada INTEGRA. El departamento de informática creó entonces una serie de vistas internas en INTEGRA para trabajar internamente con los datos, y así tener un acceso más sencillo a los datos más usados a través de una aplicación de QlikView para aquellos departamentos que no tenían dominio del SQL para las consultas a la base de datos. Tras crear las vistas y durante el desarrollo de la

aplicación, se les comunicó que el 1 de enero de 2018 se volvería a cambiar de estructura y se pasaría a una nueva, llamada INTEGRADOS. A causa de esto, se detuvo el desarrollo de la aplicación al desconocer si les serviría tras el nuevo cambio o no.

El proyecto a desarrollar consistía en crear nuevamente las vistas en INTEGRADOS, con las vistas anteriores como base y con la intención posterior de poder unificar las nuevas vistas con las viejas. Una vez creadas estas vistas, el objetivo era desarrollar la aplicación QlikView, al igual que con las vistas, basada en lo que ya existía, y a partir de ahí y del tiempo restante de prácticas, seguir desarrollando más esta aplicación. Este nuevo desarrollo estaba basado en reuniones con los departamentos que iban a usarla posteriormente para que nos dijeran cuales eran sus necesidades y así poder ir incluyéndolas en la aplicación hasta agotar el tiempo de las prácticas.

2.3. Explicación detallada del proyecto realizado en la empresa

2.3.1. Metodología y definición de tareas

Durante mi estancia en la empresa, salvo que José tuviera ya una reunión o no estuviera, nos reuníamos a primera hora para comentar las tareas pendientes que debía ir avanzando durante el día. Después, a última hora volvíamos a vernos para comentarle lo que había avanzado y las posibles dudas que tuviera y decidir que había que hacer al día siguiente. El motivo de hacer dos reuniones era que si durante la tarde surgía alguna cuestión urgente me lo comentaba en la primera reunión, y así era lo primero que hacía ese día y no necesitaba llamarme, venir a verme o esperar hasta el día siguiente.

Además, durante los primeros días me dedicaba más tiempo debido a que aún no entendía muchas cosas del funcionamiento del puerto. Más adelante, al empezar a trabajar en el diseño de la aplicación QlikView, desarrollaba un prototipo basado en lo que ya existía para mostrárselo al departamento correspondiente y, a partir de ahí, añadir nuevas funciones o realizar los cambios que ellos requiriesen.

La metodología de trabajo en cuanto a las vistas SQL era la siguiente:

1. Primero veíamos la vista de INTEGRA para comentar que campos eran importantes y cuales eran prescindibles.
2. Después mirábamos la estructura de tablas y las relaciones de INTEGRADOS para ver de dónde debía obtener los datos.

3. A continuación, por mi cuenta, buscaba todos los campos y creaba la vista en la base de datos de pruebas a la que tenía acceso.
4. Chequeábamos los principales campos de la vista con los datos reales, y si ésta era correcta, se pasaba a la base de datos real.
5. Finalmente, se empezaba a usar esa vista, y si se detectaba algún campo erróneo o faltante, se nos avisaba y volvíamos a modificar la vista.

Las vistas creadas se adjuntan en el Anexo. A continuación se describen las principales consideraciones a tener en cuenta respecto a como están hechas.

Las vistas nombradas APC_* son las que obtienen los campos requeridos para hacer las uniones con las vistas de INTEGRA. En ellas están hechos los cambios necesarios para que el tipo de datos coincida con las vistas originales, pues para poder usar un *UNION ALL* entre dos vistas los campos a unir deben ser del mismo tipo ya sea *varchar*, *number*, *date*, etc.

En estas vistas también pueden aparecer algunos campos que luego no se usan, ya que son campos internos de INTEGRADOS que cargo para comprobar que no se duplican registros y que tampoco me dejaba ninguno.

Las vistas con los nombres sin APC_* son las que unen ambas bases de datos y hay algunos campos que al ser id específicos de INTEGRA tampoco aparecerán aquí, o en otros casos si el campo no tiene su correspondiente en INTEGRADOS. Aunque se ha intentado que todos los campos importantes usados en las vistas de INTEGRA aparezcan en las nuevas vistas de INTEGRADOS. En estas vistas también aplicamos filtros para coger los datos posteriores al 1/1/2018 de INTEGRADOS y los anteriores de INTEGRA. También hay casos en los que los datos de INTEGRA se han trasladado todos correctamente a INTEGRADOS, en tal caso no existe una vista APC_* al tener todos los datos en INTEGRADOS.

Por otra parte, en algunos casos que se requería algún campo específico de INTEGRA pero que no aparecía en INTEGRADOS, se ha decidido cómo obtener ese dato o si usar otro campo de INTEGRADOS para introducirlo y luego utilizarlo.

Los *DECODES* se utilizaban generalmente para homogeneizar los posibles valores de un campo en ambas bases de datos. En las cláusulas *WHERE* se tuvieron en cuenta los campos que indican si el registro debe tenerse en cuenta o no para incluir los registros (por ejemplo, índices de matrícula repetida en los equipamientos, o índices de exención de estadísticas).

Finalmente, en casos en que un campo estaba en varias tablas se cogía el del nivel más bajo, y si éste era nulo se iba subiendo de nivel hasta encontrarlo, siempre que no se decidiera cogerlo directamente de un campo superior por requisitos de los usuarios posteriores de la vista.

Por otra parte, la metodología de trabajo en cuanto al desarrollo de la aplicación QlikView era la siguiente:

1. Preparación del prototipo basado en lo existente.
2. Reunión con el departamento correspondiente y recogida de feedback.
3. Desarrollo de nuevas funcionalidades en QlikView.
4. Repetir puntos 2 y 3 hasta quedar satisfechos con lo preparado.

Posteriormente adjunto algunas imágenes de la aplicación desarrollada, y en el Anexo se encuentra el código del script que se ejecuta en cada recarga.

2.3.2. Planificación temporal de las tareas

Planificación inicial

En un principio, el plan era de usar de mes y medio a dos meses para crear las vistas SQL en INTEGRADOS y chequearlas para que los datos fueran correctos, durante este tiempo, si tenía alguna duda y no podía contactar con José ni tenía modo de seguir avanzando con las vistas, debía ir usando el QlikView para ir descubriendo su potencial. Posteriormente debíamos usar el tiempo restante para desarrollar la aplicación en QlikView. A continuación se adjuntan los informes quincenales, donde se observa la planificación de las tareas de forma más detallada, ya que por distintas causas, ésta se tuvo que cambiar.

Informes quincenales

1ª Quincena

Eric Ferrer Escuin, Autoridad Portuaria de Castellón

- Entender el modelo de datos
- Vistas APC_Escalas y APC_Atraques
- Descubrir potencial QlikView

La primera tarea que tuve durante este tiempo fue entender el funcionamiento del puerto a rasgos generales, y más específicamente el funcionamiento de la base de datos. Para ello me encontraba cada día con mi supervisor a primera y última hora de cada día. En la primera reunión hablábamos de las tareas a realizar ese día, y en la segunda le preguntaba las dudas que me habían surgido durante el día si no eran urgentes (si lo eran le llamaba directamente).

Posteriormente empecé a crear las vistas de Escalas y Atraques, como la idea era poder unirlos a las de la base de datos anterior, tuve que crear las vistas con el prefijo APC_* en la nueva base de datos donde incluía todos los campos de dichas vistas en la base anterior para así luego poder unir ambas vistas. Después de crear las vistas me dediqué a chequearlas para comprobar que los campos mostrados eran los correctos y finalmente hacer la unión entre la base de datos anterior (INTEGRA) y la nueva (INTEGRADOS).

Paralelamente, cuando me atascaba en algún punto y hasta que mi supervisor tuviera un hueco en su agenda para ayudarme o explicarme cómo hacerlo, me dedicaba a ir experimentando con el programa QlikView para ir descubriendo todo lo que se podía hacer con él. Para ello, mi supervisor me prestó un manual de un curso que él realizó en el 2005, que aunque era de una versión más antigua, contenía todo lo básico para empezar.

2ª Quincena

Eric Ferrer Escuin, Autoridad Portuaria de Castellón

- Entender el modelo de datos
- Vista APC_Manifiestos
- Vista Expedientes
- Correcciones y chequeos de vistas anteriores
- Facturación Dominio Público

Durante este tiempo me he dedicado a seguir con los chequeos de las vistas Escalas y Atraques, ya que a medida que las iban usando se descubrían campos que no se correspondían con los correctos.

También he empezado con la siguiente vista, Manifiestos, que como en las anteriores debía hacer una versión con el prefijo APC_*. Esta vista es la más compleja de todas, ya que además de unir las dos bases de datos, primero debía unir dos sentencias SQL ya que si no se producían registros duplicados.

Paralelamente a esto, surgió una necesidad en el departamento de Dominio Público, y junto con mi supervisor decidimos aplazar el trabajo con el QlikView de Operativa Portuaria (relacionado con las vistas creadas), para priorizar esta nueva tarea. Primero hice unos Select en SQL para exportar los resultados a Excel y enviarlos al departamento y que chequeasen la facturación. Luego acordamos reunirnos con el departamento de Dominio Público para abordar sus necesidades de cara a crear las vistas para luego usarlas en QlikView.

3ª Quincena

Eric Ferrer Escuin, Autoridad Portuaria de Castellón

- Vista Dominio_Publico y FacturasDP
- Vistas Estiba y DetalleEstiba
- Revisar y modificar vistas anteriores
- Vistas necesarias para el QlikView de Dominio Público
- QlikView Dominio Público

Durante este tiempo, creé las vistas auxiliares para chequear la facturación de Dominio Público mientras creaba todas las vistas y el prototipo de QlikView.

Mientras tanto, si se descubría algún error en las otras vistas anteriores, lo corregía.

Modifiqué las vistas Estiba y DetalleEstiba para adecuarlas a los nuevos campos de la vista Manifiestos, ya que estas vistas la usaban luego en una aplicación interna del puerto y hasta este momento no les funcionaba correctamente.

Creé las vistas de Dominio Público con los campos indicados durante una reunión con ellos para luego usarlas en QlikView (vistas Actividades, Ocupaciones, Bien_Público, Cifras_Negocio, Expedientes).

También creé la vista Facturas, que luego amplié con más campos requeridos por el departamento económico-financiero, pero en este momento únicamente usé los campos que necesitaba para Dominio Público.

Empecé a crear un prototipo de QlikView para Dominio Público basándome en el que ya existía hecho para INTEGRA, y ampliándolo con nuevos datos que creía podrían ser útiles, puesto que la versión hecha para INTEGRA dejaron de desarrollarla en cuanto supieron que cambiarían de base de datos y aún era una versión básica.

4ª Quincena

Eric Ferrer Escuin, Autoridad Portuaria de Castellón

- Revisar vistas anteriores
- Prototipo QlikView Dominio Público
- Reunión Dom. Público para feedback QlikView

Como hasta ahora, si va surgiendo algún error en vistas ya creadas lo corrijo.

Tras acabar un primer prototipo de QlikView para Dominio Público realizamos una primera reunión para mostrárselo y recibir sus opiniones. Tras la reunión modifiqué el QlikView para añadir los gráficos y tablas solicitados y tras otra reunión quedó pendiente implementar unos indicadores solicitados.

5ª Quincena

Eric Ferrer Escuin, Autoridad Portuaria de Castellón

- Ultimar desarrollo QlikView Dominio Público
- Revisar incidencias con vistas anteriores
- Prototipo QlikView Operativa Portuaria
- Vistas Facturas y Líneas_Factura

Tras implementar los indicadores solicitados por el departamento de Dominio Público, empecé con el QlikView de Operativa Portuaria, para el que ya tenía las vistas creadas, e hice un primer prototipo basado en el hecho para INTEGRRA, ya que al igual que con el de Dominio Público, existía un modelo básico cuyo desarrollo se había detenido.

Paralelamente a esto, creé las vistas Facturas y Líneas_Factura, la primera ya estaba creada, pero solo tenía los campos que necesitaba en ese momento, así que la amplié con los nuevos campos necesarios para el departamento económico-financiero.

Siguiendo también mientras tanto con las correcciones puntuales de las vistas ya existentes, en su mayor parte los errores se encontraban en la vista Manifiestos, y a medida que se iban corrigiendo surgían nuevos hasta que finalmente estuvo todo correcto.

6ª Quincena

Eric Ferrer Escuin, Autoridad Portuaria de Castellón

- Prototipo QlikView Facturación
- Reunión David Castel (estadístico) para feedback QV Operativa Portuaria
- Reunión Dep. Económico para feedback QV Facturación
- Mejorar diseño de todos los QlikView
- Preparar una versión final conjunta de los QlikView



Mientras se programaba la reunión con David Castel para recibir su opinión del QlikView de Operativa Portuaria, empecé con el prototipo del QlikView de Facturación para el departamento económico-financiero.

Tras las reuniones, implementé los cambios requeridos en los QlikView y mejoré la estética del diseño para la versión definitiva de los QlikView (también el de Dominio Público).

Finalmente preparé una versión definitiva de QlikView con unos gráficos generales del puerto y posteriormente navegar a cada uno de los tres QlikView existentes.

2.3.3. Estimación de recursos del proyecto

Software utilizado

-  ORACLE SQL DEVELOPER: Es un entorno de desarrollo integrado para trabajar en código SQL con bases de datos de Oracle. Presenta una interfaz sencilla y amigable, además permite trabajar con la línea de comandos de SQL si se prefiere.
-  QLIKVIEW: Es una herramienta de inteligencia de negocios muy potente, pues permite crear aplicaciones de análisis y presentación de datos así como cuadros integrales de mando. En las figuras 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7 vemos ejemplos de la capacidad de este programa.

Lenguajes de programación utilizados

- SQL
- QLIKVIEW SCRIPT

2.3.4. Grado de consecución de los objetivos propuestos

Los objetivos propuestos eran asequibles, pues no había un objetivo final en cuanto al desarrollo de la aplicación, simplemente y a partir de ahí mejorarla todo lo posible. Por tanto, los objetivos iniciales se cumplieron todos, pues hubo tiempo de desarrollar todavía más la aplicación y añadir las nuevas solicitudes de los usuarios. A continuación muestro algunas imágenes de la aplicación final diseñada, ocultando los valores y datos, pues aún no son públicos.

En la Figura 2.1 vemos la pantalla principal de la aplicación, que es un resumen de la facturación del puerto según el tipo de tarifa, esto es por ejemplo, según si está asociado al movimiento de mercancías, a la pesca, o al uso de los terrenos portuarios. En la parte izquierda se encuentra un menú de navegación entre las distintas pantallas y en la superior derecha se indica la fecha de la última carga de datos, ya que cuando se usa con los datos reales, interesa saber cuándo se han actualizado, esto lo tendremos en todas las pantallas, ya que no todos los departamentos tienen acceso a toda la aplicación, sino solo a la parte que les corresponde. Además, en cada una de estas partes se implementan distintos filtros que el usuario puede aplicar a los datos fácilmente.

A continuación, en la Figura 2.2 vemos que el usuario puede cambiar fácilmente entre distintos tipos de gráficos que hemos definido, y así usar aquellos que más cómodos le sean, por lo que cada usuario lo verá de una forma.

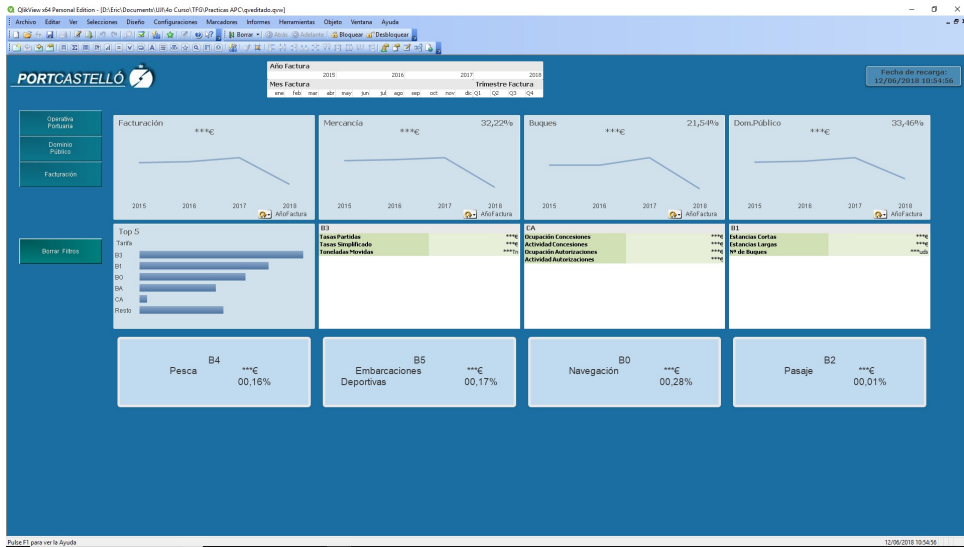


Figura 2.1: Página principal del QlikView

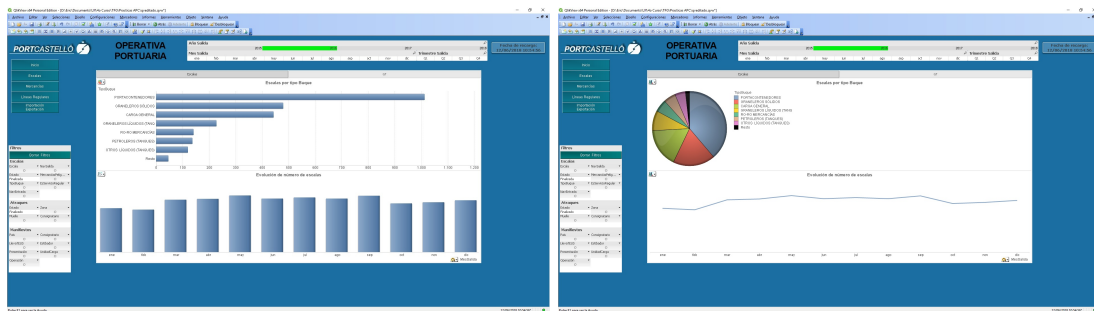


Figura 2.2: Escalas

En la Figura 2.3 vemos un ejemplo de combinación de gráficos y tablas, donde además existen varios submenús dentro que permiten visualizar estadísticas de TEUS o toneladas, o una tabla conjunta de ambas. Además, QlikView permite exportar las tablas a otros formatos para trabajar con ellas.

En la Figura 2.4 vemos una tabla pivotante, que permite agrupar o expandir las columnas y filas. En este caso vemos las mercancías movidas según su tipo, y dentro del tipo, según cómo se muevan, y si lo expandiéramos veríamos además según a que escala van asociadas. Todo ello lo vemos para cada mes.

En la Figura 2.5 vemos un ejemplo de la facilidad de filtrado de elementos, y en este caso,

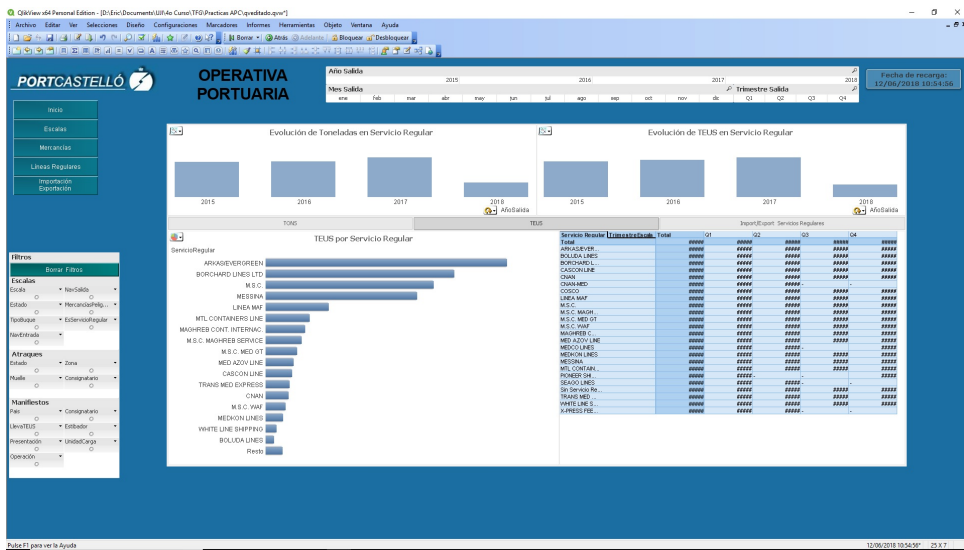


Figura 2.3: Líneas Regulares

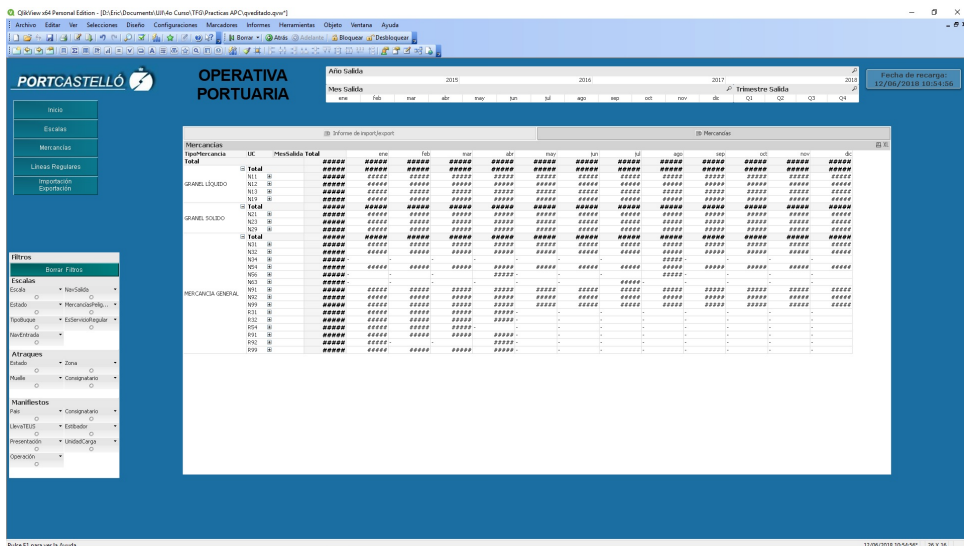


Figura 2.4: Importación/Exportación

cuando filtramos las facturas y seleccionamos una, vemos que se muestran todos los detalles de esa factura que antes estaban ocultos, pues solo se muestran cuando tenemos únicamente una factura seleccionada.

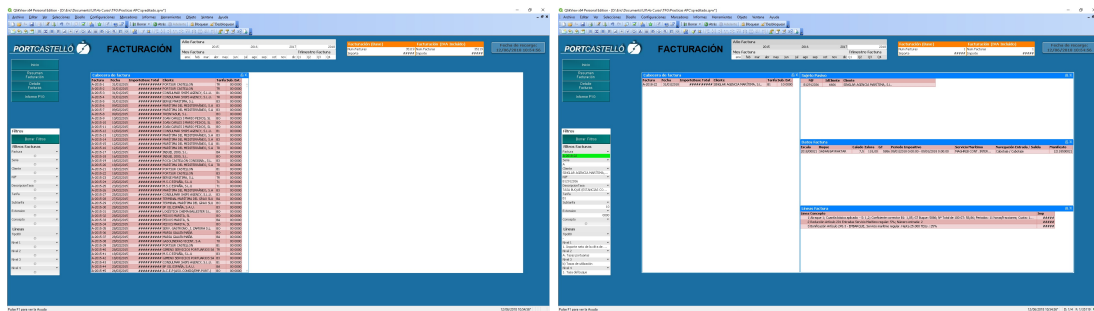


Figura 2.5: Detalle Facturas

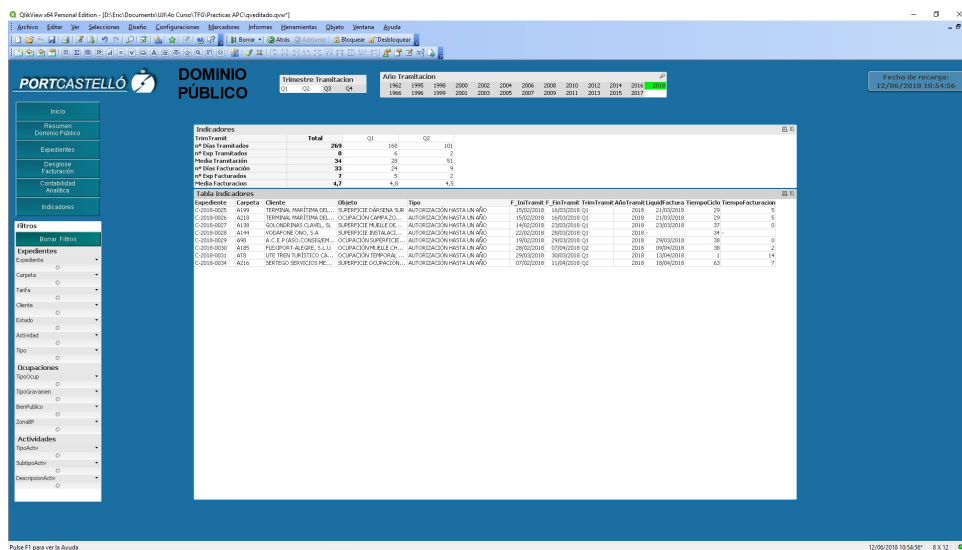


Figura 2.6: Indicadores

En la figura 2.6 vemos una de las nuevas implementaciones, y aunque hay pocos datos, pues todavía no se habían actualizado en la base de datos, vemos la capacidad de hacer cálculos del QlikView, ahorrando mucho tiempo a los usuarios, ya que antes debían hacer estos cálculos mediante *excel*.

En esta última Figura, la 2.7, vemos otra nueva función añadida para los usuarios, que el QlikView permite hacer con sencillez, ya que solo había que definir los niveles del informe PyG en el script del QlikView, y luego crear la tabla pivotante con ellos.

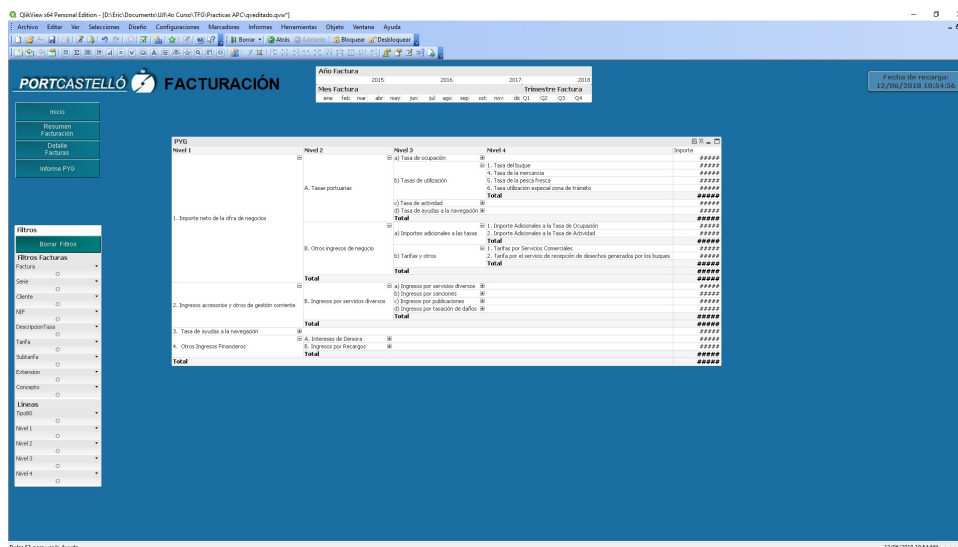


Figura 2.7: Informe PyG

2.3.5. Conclusiones

Tras terminar las prácticas, me pidieron que hiciera una presentación de la versión definitiva de la aplicación a los distintos departamentos implicados, y tras acabar, por los comentarios realizados podría decir que quedaron satisfechos con el trabajo realizado e impresionados por mi rápido aprendizaje de la terminología portuaria y el funcionamiento de éste.

Por mi parte, he terminado las prácticas agradecido a la oportunidad que me han dado, ya que hacerlas allí me ha permitido aprender a usar el programa QlikView, que era desconocido para mi, y que está en constante evolución y es muy potente. Así como también he podido aprender más de cerca el funcionamiento del puerto, ya que hay más trabajo administrativo del que puede parecer. También he de destacar el nivel de familiaridad que hay entre compañeros, pues enseguida me trataron como uno más.

Capítulo 3

Memoria TFG

3.1. Desarrollo del TFG

El trabajo va a consistir en demostrar un teorema clásico de la teoría de superficies minimales (teorema 3.1.3.5) que permite obtener ejemplos de superficies minimales partiendo de dos funciones complejas f, g que cumplan ciertos requisitos analíticos.

Para ello vamos a necesitar una serie de conceptos de variable compleja y de superficies regulares, cuyas bases se explican en asignaturas del grado, y las ampliamos un poco más a partir de las referencias bibliográficas hasta llegar a lo que necesitamos para demostrar el teorema.

Finalmente, mediante el programa *Mathematica*, dibujaremos algunas de las superficies minimales más conocidas, y también otras creadas a partir de funciones buscadas de tal forma que las forzamos a cumplir las condiciones.

3.1.1. Funciones en variable compleja

En esta sección repasaremos algunos conceptos de funciones de variable compleja. Recordemos que una función compleja, $f : \mathbb{C} \rightarrow \mathbb{C}$, es una función que tiene dominio e imagen en el cuerpo de los complejos.

Dada una función compleja $f : \mathbb{C} \rightarrow \mathbb{C}$, como todo número complejo tiene una parte real y otra imaginaria, $f(z)$ se puede escribir como $f(z) = Re(f(z)) + iIm(f(z))$ donde $Re : \mathbb{C} \rightarrow \mathbb{R}$

denota la parte real y $Im : \mathbb{C} \rightarrow \mathbb{R}$ denota la parte imaginaria.

Por tanto, existirán dos funciones $u, v : \mathbb{R}^2 \rightarrow \mathbb{R}$ tales que

$$f(z) = u(Re(z), Im(z)) + iv(Re(z), Im(z)).$$

Para simplificar la notación usamos que $z = Re(z) + iIm(z) = x + iy$ y por tanto tenemos

$$f(z) = u(x, y) + iv(x, y) \quad (3.1.1.1)$$

Veamos un ejemplo.

Ejemplo 3.1.1.1. Sea $f(z) = z^2$, entonces tenemos $f(z) = (x + iy)^2 = (x^2 + 2ixy + i^2y^2)$, de donde podemos ver que las funciones u, v serán $u(x, y) = x^2 - y^2$, $v(x, y) = 2xy$.

Usando la definición de la derivada en variable real podemos extender el concepto de derivada para funciones complejas. Dada una función $f : \mathbb{C} \rightarrow \mathbb{C}$, tenemos que la derivada en $z_0 \in \mathbb{C}$ es el límite

$$f'(z_0) = \lim_{z \rightarrow z_0} \frac{f(z) - f(z_0)}{z - z_0} \quad (3.1.1.2)$$

Si reescribimos (3.1.1.2) usando $\delta z = z - z_0$ tenemos

$$f'(z_0) = \lim_{\delta z \rightarrow 0} \frac{f(z_0 + \delta z) - f(z_0)}{\delta z} \quad (3.1.1.3)$$

Decimos que una función $f : \mathbb{C} \rightarrow \mathbb{C}$ es diferenciable en un punto $z_0 \in \mathbb{C}$ en sentido complejo si el límite en (3.1.1.3) existe. La existencia de este límite implica una fuerte restricción para las derivadas parciales de las funciones u, v conocidas como ecuaciones de Cauchy-Riemann.

Teorema 3.1.1.2 (Ver [1]). Para que una función en \mathbb{C} sea diferenciable en el punto z_0 deben cumplirse las ecuaciones de Cauchy-Riemann:

$$\begin{aligned} \frac{\partial u}{\partial x} &= \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} &= -\frac{\partial v}{\partial x} \end{aligned} \quad (3.1.1.4)$$

Demostración. Si escribimos $z_0 = x_0 + iy_0$ y $\delta z = \delta x + i\delta y$ tenemos por el Teorema 1 de la sección 12 en [1]

$$\begin{aligned} Re[f'(z_0)] &= \lim_{(\delta x, \delta y) \rightarrow (0,0)} Re \left[\frac{f(z_0 + \delta z) - f(z_0)}{\delta z} \right] \\ Im[f'(z_0)] &= \lim_{(\delta x, \delta y) \rightarrow (0,0)} Im \left[\frac{f(z_0 + \delta z) - f(z_0)}{\delta z} \right] \end{aligned} \quad (3.1.1.5)$$

donde si usamos la notación definida en (3.1.1.1) tenemos

$$\frac{f(z_0 + \delta z) - f(z_0)}{\delta z} = \frac{u(x_0 + \delta x, y_0 + \delta y) - u(x_0, y_0) + i[v(x_0 + \delta x, y_0 + \delta y) - v(x_0, y_0)]}{\delta x + i\delta y}$$

Tomando el caso particular donde $\delta z = \delta x + i0$, el punto $z_0 + \delta z$ es $(x_0 + \delta x, y_0)$ y si sustituimos en las ecuaciones de (3.1.1.5) obtenemos

$$\begin{aligned} \operatorname{Re}[f'(z_0)] &= \lim_{\delta x \rightarrow 0} \frac{u(x_0 + \delta x, y_0) - u(x_0, y_0)}{\delta x} \\ \operatorname{Im}[f'(z_0)] &= \lim_{\delta x \rightarrow 0} \frac{v(x_0 + \delta x, y_0) - v(x_0, y_0)}{\delta x} \end{aligned}$$

Y de aquí vemos

$$f'(z_0) = u_x(x_0, y_0) + iv_x(x_0, y_0) \quad (3.1.1.6)$$

Análogamente, si tomamos el caso particular $\delta z = 0 + i\delta y$ llegaremos a

$$\begin{aligned} \operatorname{Re}[f'(z_0)] &= \lim_{\delta y \rightarrow 0} \frac{v(x_0, y_0 + \delta y) - v(x_0, y_0)}{\delta y} \\ \operatorname{Im}[f'(z_0)] &= - \lim_{\delta y \rightarrow 0} \frac{u(x_0, y_0 + \delta y) - u(x_0, y_0)}{\delta y} \end{aligned}$$

De donde conseguimos

$$f'(z_0) = v_y(x_0, y_0) - iu_y(x_0, y_0) \quad (3.1.1.7)$$

Por tanto, si igualamos la parte real por un lado, y la imaginaria por otro en las ecuaciones (3.1.1.6) y (3.1.1.7) obtenemos (3.1.1.4). \diamond

En el teorema anterior hemos visto que para una función compleja $f(z) = u(x, y) + iv(x, y)$ que u y v cumplan las ecuaciones de Cauchy-Riemann es una condición necesaria para la existencia de la derivada. Veamos ahora una condición suficiente

Teorema 3.1.1.3 (Ver [1]). *Dada la función*

$$f(z) = u(x, y) + iv(x, y)$$

definida para un ϵ entorno de el punto $z_0 = x_0 + iy_0$. Supongamos que las derivadas parciales de las funciones u i v con respecto a x e y existen y son continuas en el entorno. Entonces si u y v satisfacen las ecuaciones de Cauchy-Riemann en (x_0, y_0) , la derivada $f'(z)$ existe.

Veamos ahora un ejemplo.

Ejemplo 3.1.1.4. Siguiendo con la función del ejemplo anterior, $f(z) = z^2$, sabemos que si la función estuviera en \mathbb{R} su derivada sería $f'(z) = 2z$. Veamos que de existir la derivada en \mathbb{C} , en este caso es esa la derivada. Veamos que f cumple las ecuaciones de Cauchy-Riemann (3.1.1.4). Sabemos que $u = x^2 - y^2, v = 2xy$, por tanto tendremos que

$$\begin{aligned}\frac{\partial u}{\partial x} &= 2x = 2x = \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} &= -2y = -2y = -\frac{\partial v}{\partial x}\end{aligned}$$

Como u y v tienen derivadas parciales continuas, $f'(z)$ existe para cualquier $z \in \mathbb{C}$. Usando la fórmula en (3.1.1.6) o (3.1.1.7), tenemos que $f'(z) = 2x + i2y$ que es lo mismo que $f'(z) = 2z$.

Una vez tenemos claro cuándo una función en variable compleja es diferenciable o no, podemos ver los siguientes conceptos sobre funciones.

Definición 3.1.1.5 (Ver [1]). Una función es analítica (u holomorfa) en $z_0 \in \mathbb{C}$ si su derivada existe en un entorno de z_0 . Y es analítica en un conjunto abierto $R \subset \mathbb{C}$ si lo es para cada $p \in R$. Si la función es holomorfa en todo el plano complejo, también se le llama función entera.

Definición 3.1.1.6 (Ver [1]). Se dice que una función es antiholomorfa si es la conjugada de una función holomorfa, es decir, si $f(z) = u(x, y) + iv(x, y)$ es holomorfa, $g(z) = \overline{f(z)} = u(x, y) - iv(x, y)$ es antiholomorfa.

Definición 3.1.1.7 (Ver [1]). Una función meromorfa es aquella que cumple las condiciones para ser holomorfa en todo el plano complejo salvo en un número finito de puntos aislados, que se llaman los polos de la función.

Un ejemplo de función meromorfa podría ser $f(z) = \frac{1}{z}$, ya que su derivada existen en todo \mathbb{C} salvo en el punto $z = 0$, por lo tanto hay un único punto del plano complejo (el punto $(0, 0)$) en el que no tiene derivada.

Veamos ahora la relación que existe entre funciones holomorfas y funciones armónicas. Recordemos la definición de función armónica,

Definición 3.1.1.8 (Ver [1]). Una función $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$ es armónica si tiene derivadas parciales de primer y segundo orden y además cumple la ecuación de Laplace:

$$\frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} + \cdots + \frac{\partial^2 f}{\partial x_n^2} = 0 \tag{3.1.1.8}$$

Definición 3.1.1.9 (Ver [2]). Una función $f : D \subset \mathbb{C} \rightarrow \mathbb{C}$ es armónica si las funciones que definen su parte real y su parte imaginaria son armónicas. Esto es, si expresamos f de la

siguiente forma $f(z) = u(x, y) + iv(x, y)$ y se cumple:

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= 0 \\ \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} &= 0\end{aligned}$$

Esto es pues, equivalente a que se cumpla

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$$

De la misma forma que si tenemos una función de varias variables ($f(x, y)$) y existen sus derivadas parciales ($\frac{\partial f}{\partial x}$ y $\frac{\partial f}{\partial y}$) veamos que significaría la derivada parcial respecto de z y \bar{z} en una función de variable compleja definida de la forma dada en (3.1.1.1). Veamos pues la siguiente definición.

Definición 3.1.1.10 (Ver [2]). *Las derivadas de Wirtinger se definen como sigue*

$$\begin{aligned}\frac{\partial}{\partial z} &= \frac{1}{2} \left(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right) \\ \frac{\partial}{\partial \bar{z}} &= \frac{1}{2} \left(\frac{\partial}{\partial x} + i \frac{\partial}{\partial y} \right)\end{aligned}\tag{3.1.1.9}$$

Teorema 3.1.1.11 (Ver [2]). *Dada $f(z) = u(x, y) + iv(x, y)$ de tal forma que $u, v \in C^\infty$. Las siguientes afirmaciones son equivalentes*

1. $\frac{\partial f}{\partial \bar{z}} = 0$.
2. Se cumplen las ecuaciones de Cauchy-Riemann.
3. f es holomorfa.

Demostración. Si se cumple 2, tenemos 3 por el teorema 3.1.1.2. Veamos pues ahora que 1 implica 2 y viceversa. Usando las derivadas de Wirtinger (3.1.1.9), tenemos que

$$\frac{\partial f}{\partial \bar{z}} = \frac{1}{2} \left(\frac{\partial f}{\partial x} + i \frac{\partial f}{\partial y} \right) = \frac{1}{2} \left(\frac{\partial u}{\partial x} + i \frac{\partial v}{\partial x} + i \frac{\partial u}{\partial y} + i^2 \frac{\partial v}{\partial y} \right) = \frac{1}{2} \left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) + i \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)$$

Por tanto, si se cumple 1 vemos que ambas partes, real e imaginaria deben ser 0, y obtenemos 2. Por otra parte si se cumple 2, observamos que esta derivada se anula y es 0, por lo que también tenemos 1. \diamond

Proposición 3.1.1.12. *Las derivadas de Wirtinger cumplen que:*

$$1. \frac{\partial z}{\partial z} = 1, \frac{\partial z}{\partial \bar{z}} = 0, \frac{\partial \bar{z}}{\partial z} = 0 \text{ y } \frac{\partial \bar{z}}{\partial \bar{z}} = 1$$

$$2. \Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} = 4 \frac{\partial^2}{\partial z \partial \bar{z}}$$

Demostración. Veamos 1. Sabemos que $z = x + iy$ y $\bar{z} = x - iy$

$$\frac{\partial z}{\partial z} = \frac{1}{2} \left(\frac{\partial z}{\partial x} - i \frac{\partial z}{\partial y} \right) = \frac{1}{2} (1 - i^2) = 1$$

$$\frac{\partial z}{\partial \bar{z}} = \frac{1}{2} \left(\frac{\partial z}{\partial x} + i \frac{\partial z}{\partial y} \right) = \frac{1}{2} (1 + i^2) = 0$$

$$\frac{\partial \bar{z}}{\partial z} = \frac{1}{2} \left(\frac{\partial \bar{z}}{\partial x} - i \frac{\partial \bar{z}}{\partial y} \right) = \frac{1}{2} (1 + i^2) = 0$$

$$\frac{\partial \bar{z}}{\partial \bar{z}} = \frac{1}{2} \left(\frac{\partial \bar{z}}{\partial x} + i \frac{\partial \bar{z}}{\partial y} \right) = \frac{1}{2} (1 - i^2) = 1$$

Veamos 2. Sea $f(z) = u(x, y) + iv(x, y)$

$$\begin{aligned} 4 \frac{\partial^2 f}{\partial z \partial \bar{z}} &= 4 \frac{\partial}{\partial z} \left(\frac{\partial f}{\partial \bar{z}} \right) = 4 \frac{\partial}{\partial z} \left(\frac{1}{2} \left(\frac{\partial f}{\partial x} + i \frac{\partial f}{\partial y} \right) \right) = 2 \frac{1}{2} \left(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right) \left(\frac{\partial}{\partial x} + i \frac{\partial}{\partial y} \right) = \\ &= \frac{\partial^2 f}{\partial x^2} + i \frac{\partial^2 f}{\partial x \partial y} - i \frac{\partial^2 f}{\partial y \partial x} + \frac{\partial^2 f}{\partial y^2} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \nabla^2 f = \Delta f \end{aligned}$$

◇

Estas propiedades las podemos ejemplificar como sigue

Ejemplo 3.1.1.13. Veamos esto para la función $f(z) = z^2 = (x^2 - y^2) + i(2xy)$.

$$\frac{\partial f}{\partial z} = \frac{1}{2} [(2x + i2y) - i(-2y + i2x)] = \frac{1}{2} (4x + i4y) = 2z$$

que es lo que habíamos visto antes como derivada de esta función. Por otra parte

$$\frac{\partial f}{\partial \bar{z}} = \frac{1}{2} [(2x + i2y) + i(-2y + i2x)] = 0$$

por lo que comprobamos que, como ya hemos visto anteriormente, esta función es holomorfa. Si ahora calculamos las derivadas de segundo orden de u, v , obtenemos

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 2 + (-2) = 0$$

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = 0 + 0 = 0$$

por tanto es armónica. También podríamos comprobarlo con las nuevas ecuaciones, y obtendremos que se cumplen.

Corolario 3.1.1.14 (Ver [2]). Si $f(z) = u(x, y) + iv(x, y)$ es una función en variable compleja holomorfa, entonces f es armónica.

3.1.2. Superficies regulares

Introduzcamos ahora los conceptos que usaremos de la teoría de superficies. Empecemos por el concepto de superficie,

Definición 3.1.2.1 (Ver [3]). Un subconjunto $S \subset \mathbb{R}^3$ es una superficie regular si para cada $p \in S$, existe un entorno $V \subset \mathbb{R}^3$ y una aplicación $x : (u, v) \in U \rightarrow V \cap S$ de un conjunto abierto $U \subset \mathbb{R}^2$ en $V \cap S \subset \mathbb{R}^3$ tal que:

1. x sea diferenciable
2. x sea un homeomorfismo
3. Los vectores $\frac{\partial x}{\partial u}$ y $\frac{\partial x}{\partial v}$ sean linealmente independientes

Nota. La condición 3 es equivalente en el caso de \mathbb{R}^3 a que se cumpla $\frac{\partial x}{\partial u} \times \frac{\partial x}{\partial v} \neq 0 \forall (u, v) \in U$

En este trabajo nos restringiremos al estudio de superficies parametrizadas obtenidas como $x : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ (donde x cumple 1,2,3).

Definición 3.1.2.2 (Ver [3]). Dada una superficie regular S definida por la función $X(u, v) : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$, se define el vector normal a la superficie en el punto q como el vector perpendicular unitario a S en q .

Dada la parametrización $X : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$, un vector normal N unitario viene dado por (ver [3] por ejemplo):

$$N(q) = \frac{X_u \times X_v}{|X_u \times X_v|}(q) \tag{3.1.2.1}$$

Como podemos ver por la definición del vector normal, si se cumple la condición 3 de la definición 3.1.2.1, entonces eso nos asegura que este vector existe localmente siempre si la

superficie es regular. Además, dado que este vector es perpendicular a la superficie, tenemos que los vectores X_u, X_v definen una base del plano tangente a la superficie en q . Estrechamente relacionado con el vector normal unitario podemos hablar de la orientabilidad de una superficie. Recordemos brevemente el siguiente teorema de caracterización

Teorema 3.1.2.3 (Ver [3]). *Una superficie regular $S \subset \mathbb{R}^3$ es orientable si y sólo si existe un campo de vectores normales unitarios diferenciable $N : S \rightarrow \mathbb{R}^3$. En ese caso, se dice que S es orientable con orientación N .*

El siguiente paso ahora será estudiar las propiedades métricas del plano tangente sobre una superficie por medio de la primera forma fundamental,

Definición 3.1.2.4 (Ver [3]). *Dada una superficie regular S definida por la función $X(u, v) : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$ y un vector $\vec{w} \in T_p(S)$ del plano tangente a S en p , se define la primera forma fundamental como la aplicación*

$$I_p(w) : T_p(S) \rightarrow \mathbb{R}$$

$$I_p(w) = \langle w, w \rangle_p = |w|^2 \geq 0$$

Dado un vector \vec{w} tangente a la superficie S en el punto p , tenemos que puede escribirse como vector tangente a una curva de la superficie. Esto es, existe $\alpha(t) = X(u(t), v(t))$, con $\alpha(0) = p$ y $\alpha'(0) = \vec{w}$. Si aplicamos la primera forma vemos que

$$I_p(\alpha'(0)) = \langle \alpha'(0), \alpha'(0) \rangle_p = \langle X_u u' + X_v v', X_u u' + X_v v' \rangle_p =$$

$$= \langle X_u, X_u \rangle_p (u')^2 + 2 \langle X_u, X_v \rangle_p u' v' + \langle X_v, X_v \rangle_p (v')^2$$

A los siguientes coeficientes se les llama coeficientes de la primera forma fundamental

$$E = \langle X_u, X_u \rangle_p$$

$$F = \langle X_u, X_v \rangle_p$$

$$G = \langle X_v, X_v \rangle_p$$

Y mediante estos podemos expresar en forma matricial el producto escalar entre dos vectores del plano tangente de la siguiente forma. Sean $\vec{v}, \vec{w} \in T_p(S)$ expresados en función de la base, $\{X_u, X_v\}$, del plano tangente en p , entonces

$$\langle \vec{v}, \vec{w} \rangle = \vec{v}^t \begin{pmatrix} E & F \\ F & G \end{pmatrix} \vec{w}$$

Para seguir con este repaso, vamos a ver la aplicación de Gauss para poder introducir los conceptos de curvatura gaussiana y curvatura media.

Definición 3.1.2.5 (Ver [3]). Sea $S \subset \mathbb{R}^3$ una superficie regular orientable con orientación N . La función $N : S \rightarrow \mathbb{R}^3$ que toma sus valores en la esfera unidad

$$\mathbb{S}^2 = \{(x, y, z) \in \mathbb{R}^3 / x^2 + y^2 + z^2 = 1\}$$

se denomina la función de Gauss de S y se define por tanto, como

$$N : S \rightarrow \mathbb{S}^2$$

Como vimos, el campo de vectores normales es diferenciable si la superficie es orientable. Llamaremos dN_p al diferencial de N en p :

$$dN_p : T_p(S) \rightarrow T_p(\mathbb{S}^2)$$

Como $T_p(S)$ y $T_p(\mathbb{S}^2)$ son planos paralelos, por tanto, podemos ver al diferencial de N como el endomorfismo:

$$dN_p : T_p(S) \rightarrow T_p(S).$$

Veamos ahora como se escribe el diferencial, dN_p , de N , en coordenadas locales. Sabemos que si tenemos una curva $\alpha(t)$ con $\alpha(0) = p$, su tangente es $\alpha'(t) = X_u u' + X_v v'$. Calculemos el diferencial de este vector:

$$dN_p(\alpha'(t)) = N'(u(t), v(t)) = N_u u' + N_v v'$$

Como $N_u, N_v \in T_p(S)$ los podemos escribir en función de la base del tangente $\{X_u, X_v\}$ quedando ambos así

$$\begin{aligned} N_u &= a_{11}X_u + a_{21}X_v \\ N_v &= a_{12}X_u + a_{22}X_v \end{aligned} \tag{3.1.2.2}$$

Obteniendo por tanto

$$dN(\alpha') = (a_{11}u' + a_{12}v')X_u + (a_{21}u' + a_{22}v')X_v$$

Llegando a la obtención de la expresión del diferencial de N en forma matricial

$$dN \begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} u' \\ v' \end{pmatrix} \tag{3.1.2.3}$$

Definición 3.1.2.6 (Ver [3]). Dada una superficie regular orientable S definida por la función $X(u, v) : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$ y un vector $\vec{v} \in T_p(S)$, se define la segunda forma fundamental como la aplicación

$$\begin{aligned} II_p(v) &: T_p(S) \rightarrow \mathbb{R} \\ II_p(v) &= -\langle dN_p(v), v \rangle \end{aligned}$$

Puesto que el vector \vec{v} es tangente a S en el punto p, tenemos que puede escribirse como $\alpha(t) = X(u(t), v(t))$, con $\alpha(0) = p$ y $\alpha'(0) = \vec{v}$. Si aplicamos la segunda forma vemos que

$$\begin{aligned} II_p(\alpha'(0)) &= -\langle dN(\alpha'(0)), \alpha'(0) \rangle_p = -\langle N_u u' + N_v v', X_u u' + X_v v' \rangle_p = \\ &= -\langle N_u, X_u \rangle_p (u')^2 - \langle N_u, X_v \rangle_p u'v' - \langle N_v, X_u \rangle_p u'v' - \langle N_v, X_v \rangle_p (v')^2 \end{aligned} \quad (3.1.2.4)$$

Dado que X_u, X_v son perpendiculares a N, sabemos que $\langle N, X_u \rangle = 0 = \langle N, X_v \rangle$. Derivando respecto de u obtenemos

$$\langle N_u, X_u \rangle + \langle N, X_{uu} \rangle = 0 = \langle N_u, X_v \rangle + \langle N, X_{vu} \rangle$$

y derivando respecto de v obtenemos

$$\langle N_v, X_u \rangle + \langle N, X_{uv} \rangle = 0 = \langle N_v, X_v \rangle + \langle N, X_{vv} \rangle$$

Por tanto llegamos a:

$$\begin{aligned} -\langle N_u, X_u \rangle &= \langle N, X_{uu} \rangle \\ -\langle N_u, X_v \rangle &= \langle N, X_{vu} \rangle \\ -\langle N_v, X_u \rangle &= \langle N, X_{uv} \rangle \\ -\langle N_v, X_v \rangle &= \langle N, X_{vv} \rangle \end{aligned}$$

Y dado que sabemos que $X_{vu} = X_{uv}$, vemos que $\langle N_u, X_v \rangle = \langle N_v, X_u \rangle$ Finalmente, volviendo a (3.1.2.4) llegamos a

$$II_p(\alpha'(0)) = \langle N, X_{uu} \rangle (u')^2 - 2 \langle N, X_{uv} \rangle u'v' - \langle N, X_{vv} \rangle (v')^2$$

Y de aquí obtenemos los llamados coeficientes de la segunda forma fundamental

$$\begin{aligned} e &= \langle N, X_{uu} \rangle \\ f &= \langle N, X_{uv} \rangle = \langle N, X_{vu} \rangle \\ g &= \langle N, X_{vv} \rangle \end{aligned}$$

A partir de (3.1.2.2) podemos ver que hay una relación entre dN y los coeficientes de la primera y segunda formas fundamentales.

$$\begin{aligned} -e &= -\langle N, X_{uu} \rangle = \langle N_u, X_u \rangle = \langle a_{11}X_u + a_{21}X_v, X_u \rangle = a_{11}E + a_{21}F \\ -f &= -\langle N, X_{vu} \rangle = \langle N_u, X_v \rangle = \langle a_{11}X_u + a_{21}X_v, X_v \rangle = a_{11}F + a_{21}G \\ -f &= -\langle N, X_{uv} \rangle = \langle N_v, X_u \rangle = \langle a_{12}X_u + a_{22}X_v, X_u \rangle = a_{12}E + a_{22}F \\ -g &= -\langle N, X_{vv} \rangle = \langle N_v, X_v \rangle = \langle a_{12}X_u + a_{22}X_v, X_v \rangle = a_{12}F + a_{22}G \end{aligned}$$

Y por tanto vemos que se pueden obtener los coeficientes de la matriz del diferencial de N, $a_{11}, a_{21}, a_{12}, a_{22}$, a partir de las formas fundamentales en forma matricial con la siguiente relación

$$-\begin{pmatrix} e & f \\ f & g \end{pmatrix} = \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix} \begin{pmatrix} E & F \\ F & G \end{pmatrix}$$

De donde despejando obtenemos los coeficientes

$$\begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix} = - \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1} \quad (3.1.2.5)$$

Representando $()^{-1}$ la inversa de la matriz, que es

$$\begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1} = \frac{1}{EG - F^2} \begin{pmatrix} G & -F \\ -F & E \end{pmatrix}$$

Por lo que obtenemos entonces los valores

$$\begin{aligned} a_{11} &= \frac{fF - eG}{EG - F^2} \\ a_{12} &= \frac{gF - fG}{EG - F^2} \\ a_{21} &= \frac{eF - fE}{EG - F^2} \\ a_{22} &= \frac{fF - gE}{EG - F^2} \end{aligned} \quad (3.1.2.6)$$

Veamos ahora las curvaturas de la superficie.

Definición 3.1.2.7 (Ver [3]). *Dada una superficie regular orientable S , con orientación N . Se definen las curvaturas principales k_1, k_2 como los valores propios de $-dN$. Dadas estas curvaturas principales, se definen la curvatura de Gauss, K , y la curvatura media, H , como sigue*

$$\begin{aligned} K &= k_1 \cdot k_2 \\ H &= \frac{k_1 + k_2}{2} \end{aligned} \quad (3.1.2.7)$$

De (3.1.2.7) vemos que la curvatura de Gauss se corresponde con el determinante de la matriz de dN , y de (3.1.2.5) obtenemos que su valor es

$$K = \frac{eg - f^2}{EG - F^2}$$

Por otra parte, para calcular las curvaturas principales debemos recordar que como k_1, k_2 son valores propios de la matriz, podemos obtenerlos resolviendo la ecuación $\det(A - \lambda Id) = 0$, siendo A la matriz de la que queremos obtener los valores propios y Id la matriz identidad de la dimensión de A , con λ como incógnita y sus valores los valores propios de la matriz. Por tanto vemos que se obtienen k_1, k_2 resolviendo

$$\det \begin{pmatrix} a_{11} - \lambda & a_{21} \\ a_{12} & a_{22} - \lambda \end{pmatrix} = 0$$

Vemos que queda una ecuación de segundo grado como sigue

$$\lambda^2 + (a_{11} + a_{22})\lambda + (a_{11}a_{22} - a_{12}a_{21}) = 0 \quad (3.1.2.8)$$

Volviendo a (3.1.2.5), por las definición 6 del apartado 3.2 de [3] tenemos

$$H = -\frac{1}{2} \text{Tr}(dN_p) = -\frac{1}{2} \text{Tr} \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix}$$

De donde obtenemos, sustituyendo los valores obtenidos en (3.1.2.6), que la curvatura media queda

$$H = \frac{k_1 + k_2}{2} = -\frac{a_{11} + a_{22}}{2} = \frac{1}{2} \frac{eG - 2fF + gE}{EG - F^2}$$

Como k_1, k_2 son las raíces de la ecuación (3.1.2.8), sabemos que

$$0 = (\lambda - k_1)(\lambda - k_2) = \lambda^2 - \lambda(k_1 + k_2) + k_1k_2 = \lambda^2 - 2H\lambda + K$$

De donde observamos una forma de calcular las curvaturas principales.

$$\begin{aligned} k_1 &= H + \sqrt{H^2 - K} \\ k_2 &= H - \sqrt{H^2 - K} \end{aligned}$$

En el desarrollo de la demostración del teorema principal usaremos coordenadas isotermas. Recordemos que

Definición 3.1.2.8. *Sea S una superficie parametrizada regular dada por $X : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$. Si los coeficientes de la primera forma fundamental cumplen $E = G$ y $F = 0$, entonces se dice que X es una parametrización en coordenadas isotermas de S .*

Usando coordenadas isotermas, la obtención de la curvatura media viene dada por la siguiente proposición

Proposición 3.1.2.9. *Sea $X = X(u, v)$ una superficie parametrizada regular. Si X es isoterma, entonces se cumple*

$$X_{uu} + X_{vv} = 2\lambda^2 HN \quad (3.1.2.9)$$

donde $\lambda^2 = \langle X_u, X_u \rangle = \langle X_v, X_v \rangle$

Demostración. Como X es isoterma, tenemos que $E = G$ y $F = 0$, por tanto

$$\langle X_u, X_u \rangle = \langle X_v, X_v \rangle, \quad \langle X_u, X_v \rangle = 0$$

Derivando ambas, respecto de u la primera y respecto de v la segunda obtenemos

$$\langle X_{uu}, X_u \rangle = \langle X_{vu}, X_v \rangle, \quad \langle X_{uv}, X_v \rangle = -\langle X_u, X_{vv} \rangle$$

Como $X_{uv} = X_{vu}$ y $\langle X_u, X_{vv} \rangle = \langle X_{vv}, X_u \rangle$ vemos que

$$\langle X_{uu}, X_u \rangle + \langle X_{vv}, X_u \rangle = 0$$

Por lo que

$$\langle X_{uu} + X_{vv}, X_u \rangle = 0$$

implicando esto que $X_{uu} + X_{vv}$ es perpendicular a X_u . Si ahora derivamos la primera respecto de v y la segunda respecto de u y procedemos de la misma forma llegaremos a

$$\langle X_{uu} + X_{vv}, X_v \rangle = 0$$

Por lo que $X_{uu} + X_{vv}$ también es perpendicular a X_v , de lo que se deduce que $X_{uu} + X_{vv}$ es paralelo al vector normal N . Dado que X es isoterma, tenemos que

$$H = \frac{1}{2} \frac{eG - 2fF + gE}{EG - F^2} = \frac{1}{2} \frac{e + g}{E} = \frac{1}{2} \frac{e + g}{\langle X_u, X_u \rangle}$$

Como se define $\lambda^2 = \langle X_u, X_u \rangle$, llegamos a

$$2\lambda^2 H = e + g = \langle N, X_{uu} + X_{vv} \rangle$$

Si ahora multiplicamos por el vector normal a ambos lados, pues tenemos un escalar, pasamos a tener

$$2\lambda^2 H N = \langle N, X_{uu} + X_{vv} \rangle N$$

Puesto que son paralelos y el vector normal es unitario y junto a X_u, X_v forma una base de \mathbb{R}^3 , tenemos que $\langle N, X_{uu} + X_{vv} \rangle N = X_{uu} + X_{vv}$ y sustituyendo en la expresión anterior llegamos a

$$2\lambda^2 H N = X_{uu} + X_{vv}$$

◇

3.1.3. Superficies minimales y su obtención por la parametrización de Weierstrass

El objeto de estudio de este trabajo son las superficies minimales. Recordemos que

Definición 3.1.3.1 (Ver [3]). *Sea S una superficie regular, si su curvatura media se anula idénticamente, $H(p) = 0, \forall p \in S$, a esta superficie se la denomina superficie minimal.*

Las superficies minimales están fuertemente relacionadas con las funciones armónicas por el siguiente corolario a consecuencia de la proposición 3.1.2.9,

Corolario 3.1.3.2. *Sea $X : D \subset \mathbb{C} \rightarrow \mathbb{R}^3$ una función en variable compleja supongamos que X es armónica e isoterma. Entonces la superficie parametrizada definida por X es minimal.*

Demostración. Si $X(z)$ es armónica, cumple que $X_{uu} + X_{vv} = 0$, y por 3.1.2.9 tenemos que $H = 0$, por tanto, $X : D \subset \mathbb{C} \rightarrow \mathbb{R}^3$ es minimal. \diamond

Las parametrizaciones isotermas tienen una descripción elegante en términos de funciones complejas dadas por la siguiente lema

Lema 3.1.3.3 (Ver [4]). *Sea $X : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ la parametrización de una superficie regular. Definamos las siguientes funciones complejas*

$$\phi_k(\zeta) := \frac{\partial X_k(u, v)}{\partial u} - i \frac{\partial X_k(u, v)}{\partial v}, \quad \zeta = u + iv, \quad k = 1, 2, 3.$$

Entonces si

$$\phi_1^2 + \phi_2^2 + \phi_3^2 = 0,$$

la parametrización X es isoterma.

Demostración. Observemos que

$$\begin{aligned} \phi_1^2 + \phi_2^2 + \phi_3^2 &= \sum_{k=1}^3 \left(\frac{\partial X_k(u, v)}{\partial u} \right)^2 - \sum_{k=1}^3 \left(\frac{\partial X_k(u, v)}{\partial v} \right)^2 - 2i \sum_{k=1}^3 \frac{\partial X_k(u, v)}{\partial u} \frac{\partial X_k(u, v)}{\partial v} \\ &= \langle X_u, X_u \rangle - \langle X_v, X_v \rangle - 2i \langle X_u, X_v \rangle \\ &= E - G - 2iF. \end{aligned}$$

Por lo que si $\phi_1^2 + \phi_2^2 + \phi_3^2 = 0$, tenemos que $E = G$ y que $F = 0$. Esto implica que X es una parametrización isoterma. \diamond

Veamos el siguiente lema técnico que será de ayuda a la hora de construir ejemplos,

Lema 3.1.3.4 (Ver [4]). *Sea D un dominio en el plano complejo, g una función meromorfa arbitraria en D y f una función analítica en D que tenga la propiedad de que en cada punto en el que g tiene un polo de orden m , f tiene un cero de, como mínimo, orden $2m$. Entonces las funciones*

$$\phi_1 = \frac{1}{2}f(1 - g^2), \quad \phi_2 = \frac{i}{2}f(1 + g^2), \quad \phi_3 = fg \quad (3.1.3.1)$$

serán analíticas en D y satisfacen que

$$\phi_1^2 + \phi_2^2 + \phi_3^2 = 0 \quad (3.1.3.2)$$

A la inversa, cada tripleta de funciones que cumplan (3.1.3.2) pueden ser representadas de la forma descrita en (3.1.3.1), salvo aquellas de la forma $\phi_1 \equiv i\phi_2$, $\phi_3 \equiv 0$

Demostración. Que las funciones de (3.1.3.1) cumplan (3.1.3.2) es un cálculo directo. Dada la solución a (3.1.3.2), definimos f, g así

$$f = \phi_1 - i\phi_2, \quad g = \frac{\phi_3}{\phi_1 - i\phi_2} \quad (3.1.3.3)$$

Reescribiendo (3.1.3.2) de la siguiente forma

$$(\phi_1 - i\phi_2)(\phi_1 + i\phi_2) = -\phi_3^2 \quad (3.1.3.4)$$

y operando obtenemos

$$\phi_1 + i\phi_2 = -\frac{\phi_3^2}{\phi_1 - i\phi_2} = -fg^2 \quad (3.1.3.5)$$

Si ahora operamos a partir de (3.1.3.3) y de (3.1.3.5), llegaremos a (3.1.3.1). Por tanto, vemos que las condiciones relativas a los ceros y los polos de f, g deben cumplirse, pues de otro modo, por la ecuación (3.1.3.5), $\phi_1 + i\phi_2$ no sería analítica.

Aunque hallamos escrito ϕ_i para simplificar la notación, estas funciones se encuentran en un dominio, D , del plano complejo. Por tanto vemos que esta representación solo falla en el caso de que el denominador de la expresión de g en (3.1.3.3) se anule siempre $\forall z \in D$, pero entonces vemos que despejando en (3.1.3.4), tendríamos $\phi_3 = 0$, que es justamente el caso excepcional que ya se había mencionado. \diamond

Con esto tenemos todos los ingredientes para enunciar y demostrar el teorema principal de este trabajo

Teorema 3.1.3.5. *Sea D un dominio en el plano complejo, g una función meromorfa arbitraria en D y f una función analítica en D que tenga la propiedad de que en cada punto en el que g tiene un polo de orden m , f tiene un cero de, como mínimo, orden $2m$. Entonces la superficie parametrizada por*

$$x : D \rightarrow \mathbb{R}^3, \quad x_k(z) = \operatorname{Re} \left\{ \int_{p_0}^z \phi_k(\zeta) d\zeta \right\}, \quad p_0 \in D, \quad k = 1, 2, 3, \quad (3.1.3.6)$$

es una superficie minimal. Donde ϕ_k viene definida por

$$\phi_1 = \frac{1}{2}f(1 - g^2), \quad \phi_2 = \frac{i}{2}f(1 + g^2), \quad \phi_3 = fg$$

la integral de (3.1.3.6) siendo evaluada por un camino arbitrario entre p_0 y el punto $z \in D$. Además la superficie será regular si y sólo si f satisface la condición de que se anula únicamente en los polos de g , y el orden del 0 es exactamente el doble del orden del polo de g .

Demostración. Observemos primeramente que,

$$\begin{aligned} \frac{\partial x_k}{\partial u} - i \frac{\partial x_k}{\partial v} &= \frac{\partial}{\partial u} \operatorname{Re} \left\{ \int_{p_0}^z \phi_k(\zeta) d\zeta \right\} - i \frac{\partial}{\partial v} \operatorname{Re} \left\{ \int_{p_0}^z \phi_k(\zeta) d\zeta \right\} \\ &= \operatorname{Re} \left\{ \frac{\partial z}{\partial u} \frac{d}{dz} \int_{p_0}^z \phi_k(\zeta) d\zeta \right\} - i \operatorname{Re} \left\{ \frac{\partial z}{\partial v} \frac{d}{dz} \int_{p_0}^z \phi_k(\zeta) d\zeta \right\} \\ &= \operatorname{Re}(\phi_k) - i \operatorname{Re}(i\phi_k) = \operatorname{Re}(\phi_k) - i(-\operatorname{Im}(\phi_k)) = \operatorname{Re}(\phi_k) + i\operatorname{Im}(\phi_k) \\ &= \phi_k. \end{aligned}$$

Por el lema 3.1.3.4 se cumplirá que

$$\phi_1^2 + \phi_2^2 + \phi_3^2 = 0.$$

Por tanto por el lema 3.1.3.3 la parametrización

$$x : D \rightarrow \mathbb{R}^3$$

es isoterma. Además por el lema 3.1.3.4 las funciones ϕ_1, ϕ_2, ϕ_3 son holomorfas, por tanto, al integrar sin añadir el conjugado, seguirán siendo holomorfas y por el corolario 3.1.1.14, éstas integrales serán armónicas. Por la definición de función compleja armónica, sabemos que tanto su parte real como su parte imaginaria son armónicas.

Por tanto, vemos que las coordenadas x_k son armónicas e isotermas en $z = u + iv$, con lo que por el corolario 3.1.3.2 la superficie parametrizada por $x : D \rightarrow \mathbb{R}^3$ es una superficie minimal. \diamond

3.1.4. Construcción de ejemplos de superficies minimales

A continuación vamos a hacer un programa para representar superficies minimales usando el teorema visto. Primero representamos tres superficies conocidas, el catenoide, el helicoides y la superficie de Enneper, ya que en [5] nos da la g y nuestra f la podemos obtener a partir de su dh mediante el siguiente cambio de variable:

$$f = \frac{dh}{g dz}$$

Posteriormente, conociendo las condiciones que deben cumplir f, g , inventando una g arbitraria, calcularemos una f también arbitraria que cumpla los requisitos y dibujamos la superficie.

Para el catenoide tenemos

$$f(z) = \frac{1}{z^2}$$
$$g(z) = z$$

y lo vemos representado en la Figura 3.1.

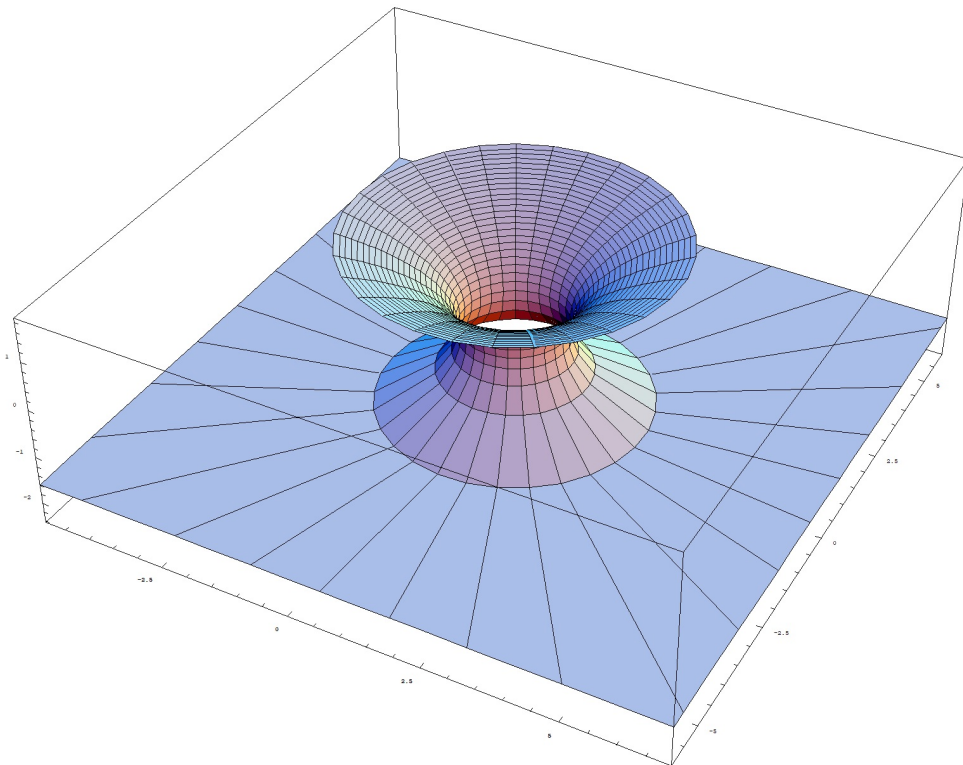


Figura 3.1: Catenoide

Para el helicoides tenemos

$$f(z) = e^z$$
$$g(z) = ie^{-z}$$

y lo vemos representado en la Figura 3.2.

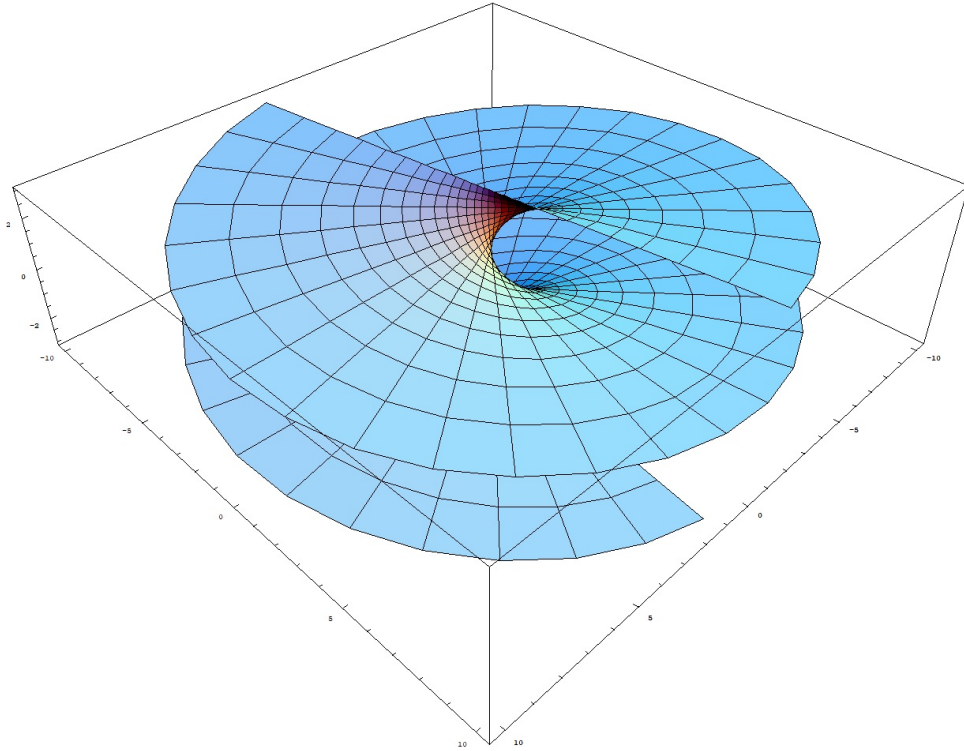


Figura 3.2: Helicoide

Para la superficie de Enneper tenemos

$$f(z) = 1$$

$$g(z) = z$$

y la representamos en la Figura 3.3.

Para la primera superficie de nuestra invención hemos definido las funciones como sigue

$$f(z) = z^2$$

$$g(z) = \frac{1}{z}$$

y la representamos en la Figura 3.4, vista desde cada uno de los ejes (X,Y,Z) respectivamente.

Para la segunda superficie de nuestra invención hemos definido las siguiente funciones

$$f(z) = (z - 1)^2(z - 2)^2$$

$$g(z) = \frac{1}{z - 1} + \frac{1}{z - 2}$$

y la representamos en la Figura 3.5.

Finalmente, adjunto el código usado para las representaciones

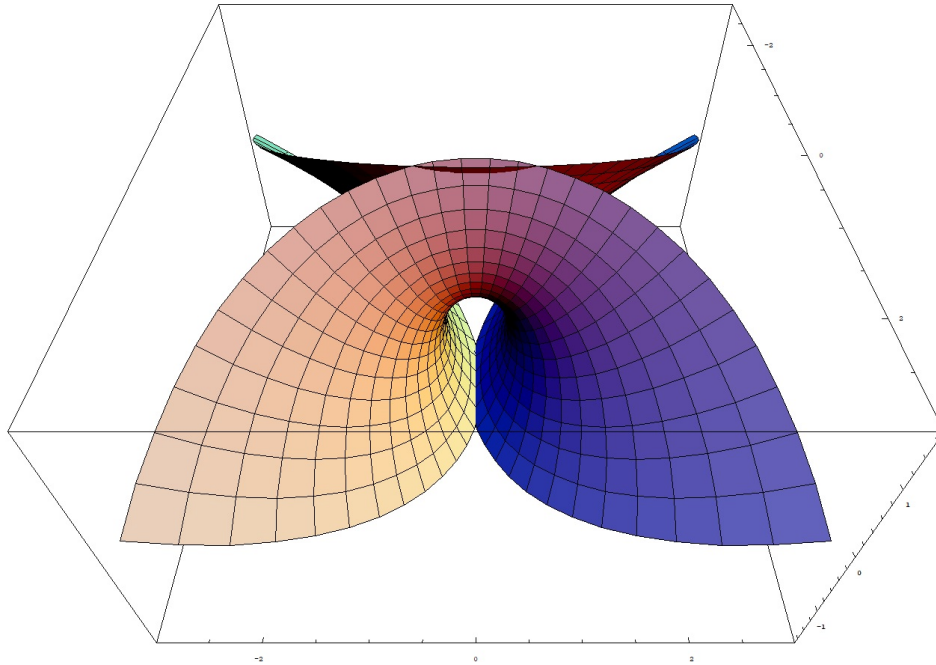


Figura 3.3: Superficie de Enneper

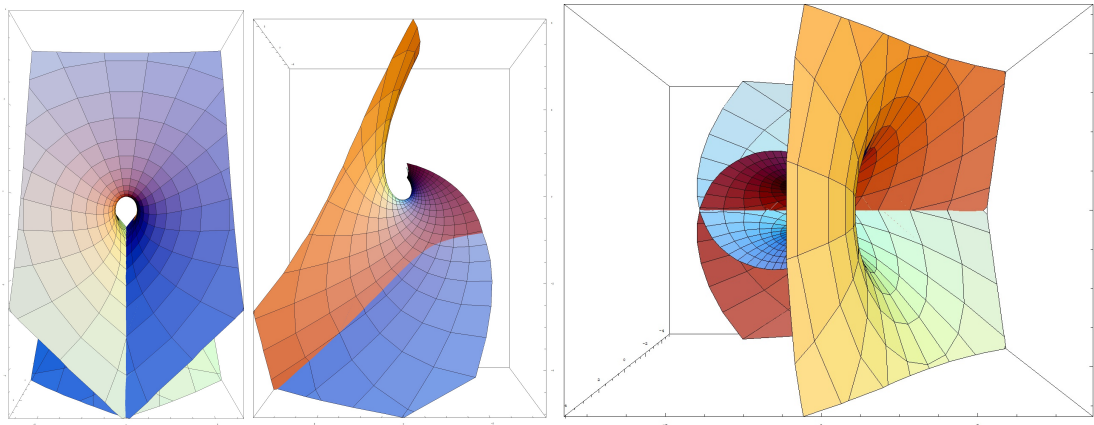


Figura 3.4: 1ª Superficie nueva

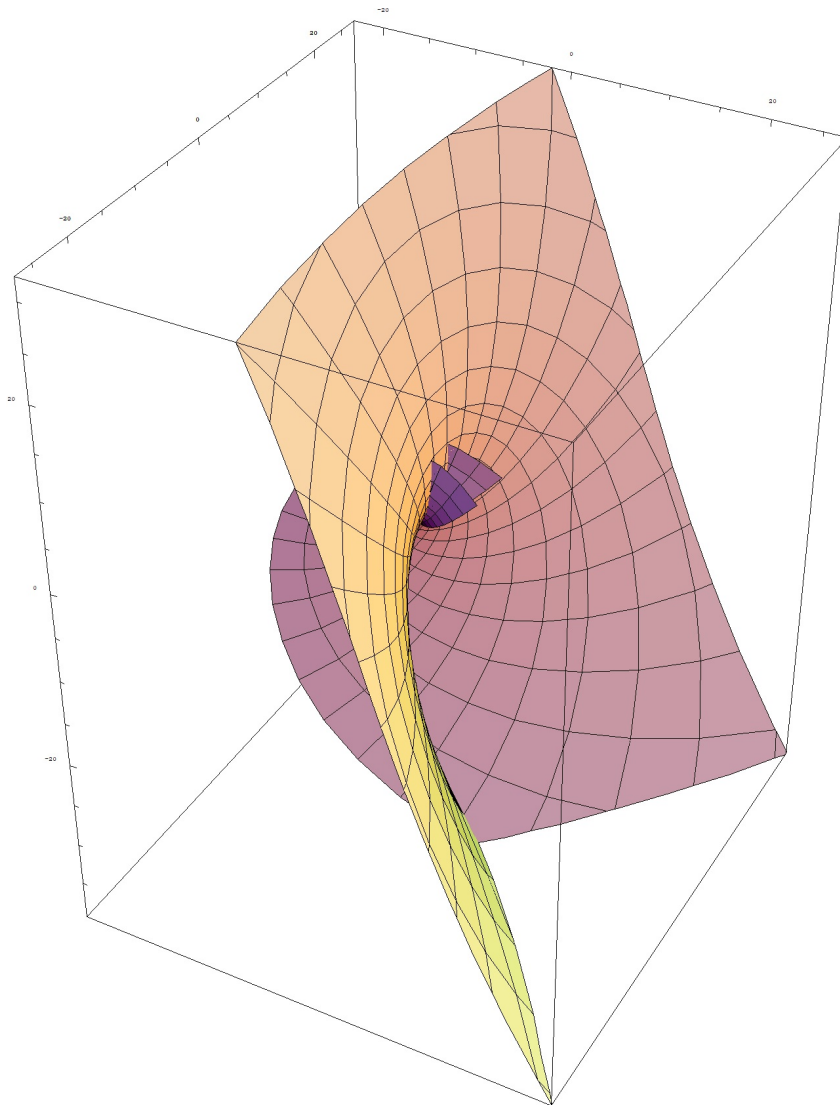


Figura 3.5: 2ª Superficie nueva

Helicoide

```
g[z_] := e^z;  
f[z_] := I * e^(-z)
```

Catenoide

```
g[z_] := z  
f[z_] := 1 / (z^2)
```

Enneper

```
g[z_] := z;  
f[z_] := 1
```

Eric 1

```
g[z_] := 1 / z;  
f[z_] := z^2 / e^z
```

Eric 2

```
g[z_] := 1 / (z - 1) + 1 / (z - 2);  
f[z_] := (z - 1)^2 * (z - 2)^2
```

```
X[u_, v_, p0_] [f_] [g_] := {Re[ $\int_{p0}^{u+i*v} f[z] * (1 - g[z]^2) / 2 dz$ ],  
  Re[ $\int_{p0}^{u+i*v} i * f[z] * (1 + g[z]^2) / 2 dz$ ], Re[ $\int_{p0}^{u+i*v} f[z] * g[z] dz$ ]}
```

```
ParametricPlot3D[X[u, v, 1] [f] [g], {u, -2., 2.}, {v, -2., 2.}]
```


Capítulo 4

Conclusiones

El teorema principal nos permite obtener ejemplos de parametrizaciones de superficies minimales, así como dibujar fácilmente superficies minimales de todo tipo. En este trabajo hemos abordado su demostración de forma auto-contenida.

Como conclusión podemos constatar que las matemáticas no están formadas por compartimentos estancos, ya que como se puede ver en la demostración del teorema principal, se basa en variable compleja aunque estemos tratando un teorema de superficies minimales.

A esta belleza teórica se une el hecho de poder representar las superficies minimales más conocidas, como el catenoide o el helicoides, y otras de nuestra invención que pueden ser todo lo complejas y raras que queramos. Aunque hay que destacar que por la forma de representarlas, tal vez no podamos verlas completas ya que dependerá de la potencia del ordenador a la hora de dibujarlas, y en algunos casos no podremos calcular la integral, que aparece en el teorema principal, en todo el dominio por este motivo.

Bibliografía

- [1] Ruel V. Churchill - Complex variables and applications (1974, McGraw-Hill). ISBN: 0-07-010855-2 (Citado en páginas 10, 28, 29 y 30.)
- [2] Walter Rudin - Real and complex analysis (1987, McGraw-Hill). ISBN: 0-07-054234-1 (Citado en páginas 10, 30, 31 y 33.)
- [3] Manfredo P. do Carmo - Differential geometry of curves and surfaces (1976, Prentice-Hall). ISBN: 0-13-212589-7 (Citado en páginas 10, 33, 34, 35, 37, 38 y 39.)
- [4] Robert Osserman - A survey of minimal surfaces (1986, Dover Publications). ISBN: 0-486-49514-0 (Citado en páginas 9 y 40.)
- [5] William H. Meeks III, Joaquín Pérez - The classical theory of minimal surfaces. Bulletin of the American Mathematical Society (2011, Vol. 48, Núm. 3, pág. 325-407) (Citado en páginas 10 y 42.)
- [6] Joaquín Pérez - Una nueva edad de oro de las superficies minimales. La Gaceta de la Real Sociedad Matemática Española (2017, Vol. 20, Núm. 1,) (Citado en página 9.)

Anexo A

Anexo I

A.1. Código SQL

A.1.1. Vista APC_ESCALAS

```
CREATE OR REPLACE FORCE VIEW "INTEGRADOSCASTELLON"."APC_ESCALAS" ("  
    EXPEDIENTE", "ANYO", "NESCALA", "FINICIO", "FSALIDA", "INICIO", "  
    SALIDA", "ESTADO", "MMPP", "NOM_PAIS_ORIGEN", "COD_PAIS_ORIGEN",  
    "COD_PUERTO_ORIGEN", "NOM_PUERTO_ORIGEN", "COD_PAIS_DESTINO", "  
    NOM_PAIS_DESTINO", "COD_PUERTO_DESTINO", "NOM_PUERTO_DESTINO", "  
    NAV_ENTRADA", "NAV_SALIDA", "COD_SERVICIO", "NOM_SERVICIO", "  
    BUQUE_ID", "COD_IMO", "BUQUE", "COD_TIPOBUQUE", "NOM_TIPOBUQUE",  
    "ESLORA", "GT", "CALADO", "MANGA", "BANDERA") AS SELECT  
E.ANYO || '/' || LPAD(TO_CHAR(E.CODESCALA),5,'0') AS EXPEDIENTE,  
TO_CHAR(E.ANYO) AS ANYO,  
TO_CHAR(E.CODESCALA) AS NESCALA,  
TO_CHAR (E.FECINIREAL, 'DD/MM/YYYY') AS FINICIO,  
TO_CHAR (E.FECFINREAL, 'DD/MM/YYYY') AS FSALIDA,  
E.FECINIREAL AS INICIO,  
E.FECFINREAL AS SALIDA,  
E.CODESTADO AS ESTADO,  
DECODE (E.INDEMMPPABORDO,1,'S','N') AS MMPP,  
PO.NOMBRE AS NOM_PAIS_ORIGEN,  
E.CODPAIORIGEN AS COD_PAIS_ORIGEN,  
E.CODPUEORIGEN AS COD_PUERTO_ORIGEN,  
PRO.NOMBRE AS NOM_PUERTO_ORIGEN,
```

```

E.CODPAIDESTINO AS COD_PAIS_DESTINO,
PD.NOMBRE AS NOM_PAIS_DESTINO,
E.CODPUEDESTINO AS COD_PUERTO_DESTINO,
PRD.NOMBRE AS NOM_PUERTO_DESTINO,
E.ESTCODTIPNAVENT AS NAV_ENTRADA ,
E.ESTCODTIPNAVSAL AS NAV_SALIDA,
E.CODSERVMARIT AS COD_SERVICIO,
SM.NOMBRE AS NOM_SERVICIO,
E.CODBUQUE AS BUQUE_ID,
BU.CODIMO AS COD_IMO,
BV.NOMBRE AS BUQUE,
BV.CODTIPOBUQUE AS COD_TIPOBUQUE,
T.DESCRIPCION AS NOM_TIPOBUQUE,
BV.ESLORATOTAL AS ESLORA,
BV.GT AS GT,
BV.CALMAXIMO AS CALADO,
BV.MANGA AS MANGA,
P.NOMBRE AS BANDERA
FROM
ESCALA E
    LEFT JOIN MBUQUE BU
        ON (BU.CODBUQUE = E.CODBUQUE)
    LEFT JOIN MBUQUEVERSION BV
        ON (BV.CODBUQUE = BU.CODBUQUE
            AND BV.FECINIVIG <=
                (CASE
                    WHEN E.FECINIREAL IS NOT NULL
                    THEN
                        TRUNC (E.FECINIREAL)
                    ELSE
                        TRUNC (E.ETA)
                    END)
            AND (BV.FECFINVIG >=
                (CASE
                    WHEN E.FECINIREAL IS NOT NULL
                    THEN
                        TRUNC (E.FECINIREAL)
                    ELSE
                        TRUNC (E.ETA)
                    END)
            OR BV.FECFINVIG IS NULL))
    LEFT JOIN TIPO_BUQUE TB

```

```

        ON BV.CODTIPOBUQUE = TB.CODTIPOBUQUE
INNER JOIN TIPO_BUQUEOPPE T
        ON (T.CODIGO = TB.CODTIPOOPPE)
INNER JOIN EST_TIPO_NAVEGACION TN
        ON (TN.CODIGO = E.ESTCODTIPNAV)
        LEFT JOIN M.SERVICIOMARITIMO SM
            ON (E.CODSERVMARIT= SM.CODSERVICIO)
INNER JOIN M.PAIS PO
        ON (E.CODPAIORIGEN = PO.CODPAIS)
INNER JOIN M.PAIS PD
        ON (E.CODPAIDESTINO = PD.CODPAIS)
INNER JOIN M.PUERTO PRO
        ON (E.CODPUEORIGEN = PRO.CODPUE AND E.CODPAIORIGEN
            = PRO.CODPAIS)
INNER JOIN M.PUERTO PRD
        ON (E.CODPUEDESTINO = PRD.CODPUE AND E.
            CODPAIDESTINO = PRD.CODPAIS)
INNER JOIN M.PAIS P
        ON (BV.BANDERA = P.CODPAIS)
WHERE E.ESTINDEXENTA = '0'
AND E.INDTRAFINT = '0'
AND E.CODESTADO IN ('IN', 'FI', 'LD');

```

A.1.1.2. Vista APC_ATRAQUES

```

CREATE OR REPLACE FORCE VIEW "INTEGRADOSCASTELLON"."APC.ATRAQUES" ("
    EXPEDIENTE", "COD_ATRAQUE", "ESTAATRA", "MOMENTO", "FECHA_INICIO"
    , "HORA_INICIO", "FECHA_FIN", "HORA_FIN", "ENTRADA", "SALIDA", "
    NORAY_INICIAL", "NORAY_FINAL", "COD_TIPO_ATRAQUE", "
    NOM_TIPO_ATRAQUE", "CALADO_ENTRADA", "CALADO_SALIDA", "
    OBSERVACIONES", "TIPO_ESTANCIA", "CONSIGNATARIO", "COD_USUARIO",
    "COD_MUELLE", "NOMMUELLE", "ZONA") AS
SELECT
E.ANYO || '/' || LPAD(TO_CHAR(E.CODESCALA),5,'0') AS EXPEDIENTE,
E.CODESTADIA AS COD_ATRAQUE,
E.CODESTADO AS ESTAATRA,
TO_CHAR('FALTA') AS MOMENTO,
TO_CHAR(E.FECATRAQUERREAL , 'DD/MM/YYYY') AS FECHA_INICIO,
TO_CHAR(E.FECATRAQUERREAL , 'HH24:MI') AS HORA_INICIO,
TO_CHAR(E.FECDESATRAQUERREAL , 'DD/MM/YYYY') AS FECHA_FIN,
TO_CHAR(E.FECDESATRAQUERREAL , 'HH24:MI') AS HORA_FIN,

```

```

TO_DATE(TO_CHAR(E.FECATRAQUERREAL , 'DD-MON-YYYY_HH24:MI:SS ') , 'DD-MON
-YYYY_HH24:MI:SS ') AS ENTRADA,
TO_DATE(TO_CHAR(E.FECDESATRAQUERREAL , 'DD-MON-YYYY_HH24:MI:SS ') , 'DD-
MON-YYYY_HH24:MI:SS ') AS SALIDA,
TO_NUMBER(E.NORAYINIREAL) AS NORAY_INICIAL,
TO_NUMBER(E.NORAYFINREAL) AS NORAY_FINAL,
E.CODTIPATRAQUERREAL AS COD_TIPO_ATRAQUE,
A.DESCRIPCION AS NOM_TIPO_ATRAQUE,
E.CALADOENTRADA AS CALADO_ENTRADA,
E.CALADOSALIDA AS CALADO_SALIDA,
E.OBSERVACIONES AS OBSERVACIONES,
TO_CHAR('FALTA') AS TIPO_ESTANCIA,
E.NOMBRECONSIGESCALA AS CONSIGNATARIO,
TO_CHAR(E.IDSUJPASIVO) AS COD_USUARIO,
E.CODMUELLEINIREAL AS COD_MUELLE,
M.DESCRIPCION AS NOMMUELLE,
TO_CHAR(M.CODZONAAGUA) AS ZONA
FROM ESTADIA E
LEFT JOIN MMUELLE M
ON (E.CODMUELLEINIREAL = M.CODIGO)
LEFT JOIN FORMA_ATRAQUE A
ON (E.CODTIPATRAQUERREAL = A.CODTIPATRAQUE)
LEFT JOIN ESCALA ES
ON (E.CODESCALA=ES.CODESCALA AND E.ANYO=ES.ANYO)
WHERE ES.INDTRAFINT = '0';

```

A.1.3. Vista APC_MANIFIESTOS

```

CREATE OR REPLACE FORCE VIEW "INTEGRADOSCASTELLON"."
APC_MANIFIESTOS" ("CODSER" , "CODIRAMO" , "ESTADOMA" , "ESTADOBL" ,
"ESTADOTRAMO" , "NUMORDEN" , "ID_EQ_P" , "EQ_TC" , "EXPEDIENTE" , "
FSALIDAESCALA" , "TIPO_OPERACION" , "PUERTO" , "PAIS" , "COD_MUELLE
" , "NOMMUELLE" , "COD_ESTIBADOR" , "NOMESTIBADOR" , "
COD_CONSIGNATARIO" , "NOMCONSIGNATARIO" , "COD_SERVICIO" , "
NOM.SERVICIO" , "COD_INST_ESPECIAL" , "NOM_INST_ESPECIAL" , "
CODMERC" , "UC" , "NOMLUC" , "COD_TIPO_MERCANCIA" , "
NOM.TIPO_MERCANCIA" , "TONS" , "TEUS" , "BULTOS" , "NUMLUC") AS
SELECT DISTINCT
EQ.CODSER AS CODSER,
EQ.CODIRAMO AS CODIRAMO,
DECODE(T.REVISADO,1 , 'C' , 'I ') AS ESTADOMA,

```



```

DECODE(T.REVISADO,1,'C','I') AS ESTADOB,
T.CODESTADO AS ESTADOTRAMO,
EQ.NUMORDEN AS NUMORDEN,
TO_CHAR(EQ.CODEQUIP) AS ID_EQ_P,
EQ.TIPOCONOC AS EQ_TC,
C.ANYO || '/' || LPAD(TO_CHAR(C.CODESCALA),5,'0') AS EXPEDIENTE,
TO_CHAR(E.FECFINREAL,'DD/MM/YYYY') AS FSALIDAESCALA,
CON.ESTCODTIPOPEROPPE AS TIPO_OPERACION,
DECODE(CON.ESTCODTIPOPEROPPE,'D',CODPUE1,'E',CODPUE3,'DT',CODPUE1,'
ET',CODPUE3,'-') AS PUERTO,
DECODE(CON.ESTCODTIPOPEROPPE,'D',CODPAI1,'E',CODPAI3,'DT',CODPAI1,'
ET',CODPAI3,'-') AS PAIS,
CASE
WHEN CON.CODMUELLE IS NOT NULL THEN CON.CODMUELLE
ELSE T.CODMUELLE
END
AS CODMUELLE,
M.DESCRIPCION AS NOMMUELLE,
EST.CODAPSCT AS COD_ESTIBADOR,
EST.NOMBRECOMPLETO AS NOMESTIBADOR,
CSG.CODAPSCT AS COD_CONSIGNATARIO,
CSG.NOMBRECOMPLETO AS NOM_CONSIGNATARIO,
T.CODSERVENT AS COD_SERVICIO,
SM.NOMBRE AS NOM_SERVICIO,
P.CODINSTESP AS COD_INST_ESPECIAL,
IE.DESCRIPCION AS NOM_INST_ESPECIAL,
DECODE(EQ.CODESTADO,'4','8609*','8609T') AS COD_MERC,
EQ.ESTTUCAOPPE AS UC,
TEQ.DESCRIPCION AS NOMLUC,
TEQ.CODTIPOMERCSIGMA AS COD_TIPO_MERCANCIA,
DECODE(TEQ.CODTIPOMERCSIGMA,'MG','MERCANCIA_GENERAL','LP','GRANEL
_LIQUIDO','LO','GRANEL_LIQUIDO','SG','GRANEL_SOLIDO','OTROS'
) AS NOM_TIPO_MERCANCIA,
EQ.TARA/1000 AS TONS,
EQ.ESTNUMTEUS AS TEUS,
0 AS BULTOS,
0 AS NUMLUC
FROM G3.EQUIPAMIENTO EQ
JOIN G3.CABECERA C
ON(C.CODPUE=EQ.CODPUE AND C.ANYO=EQ.ANYO AND C.CODSER=EQ.
CODSER)
JOIN ESCALA E

```

```

        ON (C.CODPUEESCALA = E.CODPUE AND C.ANYOESCALA = E.ANYO
            AND C.CODESCALA = E.CODESCALA)
JOIN G3.CONOCIMIENTO CON
        ON(CON.CODPUE=EQ.CODPUE AND CON.ANYO=EQ.ANYO AND          CON.
            CODIRAMO=EQ.CODIRAMO AND CON.CODSER=EQ.CODSER AND      CON.
            NUMORDEN = EQ.NUMORDEN AND CON.TIPOCONOC=EQ.TIPOCONOC)
JOIN G3.TRAMO T
        ON(T.CODPUE=EQ.CODPUE AND T.ANYO=EQ.ANYO AND          T.CODIRAMO=
            EQ.CODIRAMO AND T.CODSER=EQ.CODSER)
JOIN M.USUARIO EST
        ON (C.IDESTIBADOR = EST.CODUSUARIO)
JOIN M.USUARIO CSG
        ON (C.IDCONSIGNATARIO = CSG.CODUSUARIO)
LEFT JOIN M.SERVICIOMARITIMO SM
        ON((CASE
            WHEN T.CODSERVENT IS NOT NULL THEN T.CODSERVENT
            ELSE C.CODSERVMARIT
            END)= SM.CODSERVICIO)
LEFT JOIN G3.PARTIDA P
        ON(P.CODPUE=EQ.CODPUE AND P.ANYO=EQ.ANYO AND          P.CODIRAMO=
            EQ.CODIRAMO AND P.CODSER=EQ.CODSER AND              P.NUMORDEN =
            EQ.NUMORDEN AND P.TIPOCONOC=EQ.TIPOCONOC)
LEFT JOIN M.INSTALESPECIAL IE
        ON(P.CODINSTESP=IE.CODIGO)
JOIN M.MUELLE M
        ON ((CASE
            WHEN CON.CODMUELLE IS NOT NULL THEN CON.CODMUELLE
            ELSE T.CODMUELLE
            END) = M.CODIGO)
JOIN EST.TUCAOPPE TEQ
        ON(EQ.ESTTUCAOPPE = TEQ.CODIGO)
WHERE (E.CODESTADO IN( 'FI' , 'LD' ) OR E.CODESTADO IS NULL)
AND (EQ.INDBAJA=0 OR EQ.INDBAJA IS NULL) AND EQ.INDMATREPETIDA='0'

UNION ALL

SELECT DISTINCT
P.CODSER AS CODSER,
P.CODIRAMO AS CODIRAMO,
DECODE(T.REVISADO,1 , 'C' , 'I' ) AS ESTADOMA,

```

```

DECODE(T.REVISADO,1,'C','I') AS ESTADOBL,
T.CODESTADO AS ESTADOTRAMO,
P.NUMORDEN AS NUMORDEN,
TO_CHAR(P.NUMPARTIDA) AS ID_EQ_P,
P.TIPOCONOC AS EQ_TC,
C.ANYO || '/' || LPAD(TO_CHAR(C.CODESCALA),5,'0') AS EXPEDIENTE,
TO_CHAR(E.FECFINREAL,'DD/MM/YYYY') AS FSALIDAESCALA,
CON.ESTCODTIPOPEROPPE AS TIPO_OPERACION,
DECODE(CON.ESTCODTIPOPEROPPE,'D',CODPUE1,'E',CODPUE3,'DT',CODPUE1,'
ET',CODPUE3,'-') AS PUERTO,
DECODE(CON.ESTCODTIPOPEROPPE,'D',CODPAI1,'E',CODPAI3,'DT',CODPAI1,'
ET',CODPAI3,'-') AS PAIS,
CASE
  WHEN P.CODMUELLE IS NOT NULL THEN P.CODMUELLE
  WHEN CON.CODMUELLE IS NOT NULL THEN CON.CODMUELLE
  ELSE T.CODMUELLE
END
AS CODMUELLE,
M.DESCRIPCION AS NOMMUELLE,
EST.CODAPSCT AS COD_ESTIBADOR,
EST.NOMBRECOMPLETO AS NOMESTIBADOR,
CSG.CODAPSCT AS COD_CONSIGNATARIO,
CSG.NOMBRECOMPLETO AS NOM_CONSIGNATARIO,
T.CODSERVENT AS COD_SERVICIO,
SM.NOMBRE AS NOM_SERVICIO,
P.CODINSTESP AS COD_INST_ESPECIAL,
IE.DESCRIPCION AS NOM_INST_ESPECIAL,
P.CODARAN AS COD_MERC,
MC.ESTCODTUCAOPPE AS UC,
UC.DESCRIPCION AS NOMLUC,
UC.CODTIPOMERCSIGMA AS COD_TIPO_MERCANCIA,
DECODE(UC.CODTIPOMERCSIGMA,'MG','MERCANCIA_GENERAL','LP','GRANEL_
LIQUIDO','LO','GRANEL_LIQUIDO','SG','GRANEL_SOLIDO','OTROS') AS
NOM_TIPO_MERCANCIA,
P.PESO/1000 AS TONS,
0 AS TEUS,
P.NUMBULTOS AS BULTOS,
0 AS NUMLUC

FROM G3.PARTIDA P
JOIN G3.CABECERA C

```

```

ON(C.CODPUE=P.CODPUE AND C.ANYO=P.ANYO AND C.CODSER=P.
CODSER)
JOIN ESCALA E
ON (C.CODPUEESCALA = E.CODPUE AND C.ANYOESCALA = E.ANYO
AND C.CODESCALA = E.CODESCALA)
JOIN G3.CONOCIMIENTO CON
ON(CON.CODPUE=P.CODPUE AND CON.ANYO=P.ANYO AND CON.CODTRAMO
=P.CODTRAMO AND CON.CODSER=P.CODSER AND CON.NUMORDEN
= P.NUMORDEN AND CON.TIPOCONOC=P.TIPOCONOC)
JOIN G3.TRAMO T
ON(T.CODPUE=P.CODPUE AND T.ANYO=P.ANYO AND T.CODTRAMO=P
.CODTRAMO AND T.CODSER=P.CODSER)
LEFT JOIN M.USUARIO EST
ON (CASE
WHEN P.IDESTIBADOR IS NOT NULL THEN P.IDESTIBADOR
ELSE C.IDESTIBADOR
END) = EST.CODUSUARIO
JOIN M.USUARIO CSG
ON (C.IDCONSIGNATARIO = CSG.CODUSUARIO)
LEFT JOIN M.SERVICIOMARITIMO SM
ON((CASE
WHEN T.CODSERVENT IS NOT NULL THEN T.CODSERVENT
ELSE C.CODSERVMARIT
END)= SM.CODSERVICIO)
LEFT JOIN M.INSTALESPECIAL IE
ON(P.CODINSTESP=IE.CODIGO)
JOIN M.MUELLE M
ON ((CASE
WHEN P.CODMUELLE IS NOT NULL THEN P.CODMUELLE
WHEN CON.CODMUELLE IS NOT NULL THEN CON.CODMUELLE
ELSE T.CODMUELLE
END) = M.CODIGO)
JOIN M.MERCANCIA MC
ON(MC.CODARAN=P.CODARAN)
JOIN EST.TUCAOPPE UC
ON(MC.ESTCODTUCAOPPE=UC.CODIGO)

WHERE (E.CODESTADO IN( 'FI' , 'LD' ) OR E.CODESTADO IS NULL)
AND (P.INDBAJA=0 OR P.INDBAJA IS NULL)
AND MC.ESTINDEXENTA = '0'
AND C.ESTINDEXENTA = '0'
AND T.ESTINDEXENTA = '0';

```

A.1.4. Vista ESCALAS

```
CREATE OR REPLACE VIEW "INTEGRADOSCASTELLON"."ESCALAS" ("ANYO", "
BANDERA", "BUQUE", "BUQUE.ID", "CALADO", "COD_IMO", "
COD_PAIS_DESTINO", "COD_PAIS_ORIGEN", "COD_PUERTO_DESTINO", "
COD_PUERTO_ORIGEN", "COD_TIPOBUQUE", "COD_SERVICIO", "ESLORA", "
ESTADO", "EXPEDIENTE", "FINICIO", "FSALIDA", "GT", "INICIO", "
MANGA", "MMPP", "NAV_ENTRADA", "NAV_SALIDA", "NECALA", "
NOM_PAIS_DESTINO", "NOM_PAIS_ORIGEN", "NOM_PUERTO_DESTINO", "
NOM_PUERTO_ORIGEN", "NOM_TIPOBUQUE", "NOM_SERVICIO", "SALIDA") AS
SELECT
ANYO, BANDERA, BUQUE, BUQUE.ID, CALADO, COD_IMO, COD_PAIS_DESTINO,
COD_PAIS_ORIGEN, COD_PUERTO_DESTINO, COD_PUERTO_ORIGEN,
COD_TIPOBUQUE, COD_SERVICIO, ESLORA, ESTADO, EXPEDIENTE, FINICIO,
FSALIDA, GT, INICIO, MANGA, MMPP, NAV_ENTRADA, NAV_SALIDA,
NECALA, NOM_PAIS_DESTINO, NOM_PAIS_ORIGEN, NOM_PUERTO_DESTINO,
NOM_PUERTO_ORIGEN, NOM_TIPOBUQUE, NOM_SERVICIO, SALIDA
FROM INTEGRADOSCASTELLON.APC_ESCALAS
WHERE SALIDA >= TO_DATE('01/01/2018', 'DD/MM/YYYY')

UNION ALL

SELECT
ANYO, BANDERA, BUQUE, BUQUE.ID, CALADO, COD_IMO, COD_PAIS_DESTINO,
COD_PAIS_ORIGEN, COD_PUERTO_DESTINO, COD_PUERTO_ORIGEN,
COD_TIPOBUQUE, COD_TRAFICO, ESLORA,
DECODE(ESTADO, 'F', 'FI', 'I', 'IN', 'S', 'AC', 'C', 'AN', 'A', 'AU') AS
ESTADO,
EXPEDIENTE, FINICIO, FSALIDA, GT, INICIO, MANGA, MMPP, NAV_ENTRADA,
NAV_SALIDA, NECALA, NOM_PAIS_DESTINO, NOM_PAIS_ORIGEN,
NOM_PUERTO_DESTINO, NOM_PUERTO_ORIGEN, NOM_TIPOBUQUE, NOM_TRAFICO
, SALIDA
FROM INTEGRA.ESCALAS
WHERE SALIDA < TO_DATE('01/01/2018', 'DD/MM/YYYY');
```

A.1.5. Vista ATRAQUES

```
CREATE OR REPLACE FORCE VIEW "INTEGRADOSCASTELLON"."ATRAQUES" ("
EXPEDIENTE", "COD_ATRAQUE", "ESTAATRA", "FECHA_INICIO", "
HORA_INICIO", "FECHA_FIN", "HORA_FIN", "ENTRADA", "SALIDA", "
NORAY_INICIAL", "NORAY_FINAL", "COD_TIPO_ATRAQUE", "
```

```

NOM.TIPO.ATRAQUE" , "CALADO.ENTRADA" , "CALADO.SALIDA" , "
OBSERVACIONES" , "TIPO.ESTANCIA" , "CONSIGNATARIO" , "COD.USUARIO" ,
"COD.MUELLE" , "NOM.MUELLE" ,"ZONA" ) AS
SELECT
EXPEDIENTE, COD.ATRAQUE, ESTAAATRA, FECHA.INICIO, HORA.INICIO,
FECHA.FIN, HORA.FIN, ENTRADA, SALIDA, NORAY.INICIAL, NORAY.FINAL,
COD.TIPO.ATRAQUE, NOM.TIPO.ATRAQUE, CALADO.ENTRADA,
CALADO.SALIDA, OBSERVACIONES, TIPO.ESTANCIA, CONSIGNATARIO,
COD.USUARIO, COD.MUELLE, NOM.MUELLE, ZONA
FROM APC.ATRAQUES
WHERE SALIDA >= TO_DATE( '01/01/2018' , 'DD/MM/YYYY' )

UNION ALL

SELECT
EXPEDIENTE, COD.ATRAQUE,
DECODE(ESTAAATRA, 'F' , 'FI' , 'I' , 'IN' , 'C' , 'AN' , 'A' , 'AU' , 'S' , 'AC') AS
ESTAAATRA,
FECHA.INICIO, HORA.INICIO, FECHA.FIN,
HORA.FIN, ENTRADA, SALIDA,
ATRD.NORAY.INICIAL AS NORAY.INICIAL,
ATRD.NORAY.FINAL AS NORAY.FINAL,
DECODE(COD.TIPO.ATRAQUE, 'A' , 'RXX' , 'B' , 'YXX' , 'C' , 'AXM' , 'P' , 'APX' , 'F'
, 'FXX' ) AS COD.TIPO.ATRAQUE,
DECODE(NOM.TIPO.ATRAQUE, 'ABARLOADO' , 'Abarloado.....
.....' , 'A.BOYA' , 'Amarrado_en_campo_boyas' , 'DE.COSTADO' , 'Atracado..
al_muelle.....' , 'DE.PUNTA' , 'Atracado_de_punta.....
.....' , 'FONDEADO' , 'Fondeo' ) AS NOM.TIPO.ATRAQUE,
ATRD.CALADO.ENTRADA AS CALADO.ENTRADA,
ATRD.CALADO.SALIDA AS CALADO.SALIDA,
ATRD.OBSERVACIONES AS OBSERVACIONES,
ATRD.TIPO.ESTANCIA AS TIPO.ESTANCIA,
CONSIGNATARIO, COD.USUARIO, COD.MUELLE, NOM.MUELLE, ZONA
FROM INTEGRA.ATRAQUES
WHERE SALIDA < TO_DATE( '01/01/2018' , 'DD/MM/YYYY' );

```

A.1.6. Vista MANIFIESTOS

```

CREATE OR REPLACE FORCE VIEW "INTEGRADOSCASTELLON"."MANIFIESTOS" (
EXPEDIENTE" , "ESTADOMA" , "ESTADOBL" , "FSALIDAESCALA" , "
COD.OPERACION" , "PUERTO" , "PAIS" , "COD.MUELLE" , "NOM.MUELLE" , "

```

```

COD_ESTIBADOR" , "NOM_ESTIBADOR" , "COD_CONSIGNATARIO" , "
NOM_CONSIGNATARIO" , "COD_SERVICIO" , "NOM_SERVICIO" , "
COD_INST_ESPECIAL" , "NOM_INST_ESPECIAL" , "COD_MERC" , "UC" , "
NOM_LUC" , "COD_TIPO_MERCANCIA" , "NOM_TIPO_MERCANCIA" , "TONS" , "
TEUS" , "BULTOS" , "NUM_LUC" ) AS
SELECT
EXPEDIENTE, ESTADOMA, ESTADOB, FSALIDAESCALA, TIPO_OPERACION,
PUERTO, PAIS, COD_MUELLE, NOM_MUELLE, COD_ESTIBADOR,
NOM_ESTIBADOR, COD_CONSIGNATARIO, NOM_CONSIGNATARIO, COD_SERVICIO
, NOM_SERVICIO, COD_INST_ESPECIAL, NOM_INST_ESPECIAL, COD_MERC,
UC, NOM_LUC, COD_TIPO_MERCANCIA, NOM_TIPO_MERCANCIA, TONS, TEUS,
BULTOS, NUM_LUC
FROM INTEGRADOSCASTELLON.APC_MANIFIESTOS
WHERE TO_DATE(FSALIDAESCALA, 'DD/MM/YYYY') >= TO_DATE( '01/01/2018' , '
DD/MM/YYYY' )

UNION ALL

SELECT
EXPEDIENTE, ESTADOMA, ESTADOB, FSALIDAESCALA, COD_OPERACION, PUERTO
, PAIS, COD_MUELLE, NOM_MUELLE, COD_ESTIBADOR, NOM_ESTIBADOR,
COD_CONSIGNATARIO, NOM_CONSIGNATARIO, COD_SERVICIO, NOM_SERVICIO,
COD_INST_ESPECIAL, NOM_INST_ESPECIAL, COD_MERC, UC, NOM_LUC,
COD_TIPO_MERCANCIA, NOM_TIPO_MERCANCIA, TONS, TEUS, BULTOS,
NUM_LUC
FROM INTEGRA.MANIFIESTOS
WHERE TO_DATE(FSALIDAESCALA, 'DD/MM/YYYY') < TO_DATE( '01/01/2018' , 'DD
/MM/YYYY' ) ;

```

A.1.7. Vista FACTURAS_DP y código auxiliar

```

CREATE OR REPLACE FORCE VIEW "INTEGRADOSCASTELLON"."FACTURAS_DP" (
"CODFAC" ,"NOMBRECLIENTE" ,"CODBOR" ,"CODDP" , "EXPEDIENTE" , "SERIE" ,
"FECHA" , "IMPNET" ,"IMPTOT" , "IDASPECTOTARIFA" , "
IDASPECTOSUBTARIFA" , "ACTLOCUP" , "PERIODO" ) AS
SELECT
B.numfacemit ,
B.NOMBRECLIENTE,
B.CODBORRADOR,
B.CODDP,
XP.EXPEDIENTE,

```

```

B.CODSERFACTURA,
B.FECLIQ,
b.impnet,
B.IMPTOT,
B.IDASPECTOTARIFA,
B.IDASPECTOSUBTARIFA,
DECODE(B.IDASPECTOSUBTARIFA, '30', 'BO', '40', 'BA', '50', 'BO', '60', 'BA',
NULL) AS ACTLOCUP,
CASE
WHEN B.IDASPECTOTARIFA='TA' THEN XP.PERIODOLIQACTIV
WHEN B.IDASPECTOTARIFA='TO' THEN XP.PERIODOLIQCUP
WHEN DECODE(B.IDASPECTOSUBTARIFA, '30', 'BO', '50', 'BO', NULL)='BO'
THEN XP.PERIODOLIQCUP
WHEN DECODE(B.IDASPECTOSUBTARIFA, '40', 'BA', '60', 'BA', NULL)='BA'
THEN XP.PERIODOLIQACTIV
ELSE NULL
END AS PERIODO
FROM LQ.BORRADOR B
JOIN DP.EXPEDIENTE XP
ON(B.CODDP=XP.CODIGO)
WHERE B.NUMFACMIT IS NOT NULL
ORDER BY 1;

```

```

SELECT
SUM(IMPNET) ,
SUM(IMPTOT)
FROM FACTURAS.DP
WHERE IDASPECTOTARIFA='CA' OR IDASPECTOTARIFA='TO' OR
IDASPECTOTARIFA='TA'
;

SELECT
SERIE,
CODFAC,
CODBOR,
FECHA,
NOMBRECLIENTE,
SUM(IMPNET) ,
SUM(IMPTOT) ,
IDASPECTOTARIFA,
IDASPECTOSUBTARIFA,
PERIODO

```



```

FROM FACTURAS.DP
WHERE IDASPECTOTARIFA='CA' OR IDASPECTOTARIFA='TO' OR
    IDASPECTOTARIFA='TA'
GROUP BY CODFAC, codbor ,SERIE ,NOMBRECLIENTE,FECHA,IDASPECTOTARIFA,
    IDASPECTOSUBTARIFA,PERIODO
ORDER BY IDASPECTOTARIFA,IDASPECTOSUBTARIFA,codfac desc
;

```

A.1.8. Vista EXPEDIENTES

```

CREATE OR REPLACE FORCE VIEW "INTEGRADOSCASTELLON"."EXPEDIENTES" (
    "EXPEDIENTE", "OBJETO", "NOMBRECLIENTE", "TIPO_EXP", "ESTADO", "
    CARPETA", "ACTIVIDAD_PRINCIPAL", "IVA_OCUP", "IVA_ACT", "
    PERIODO_OCUP", "PERIODO_ACT", "F_OTORGAMIENTO", "F_RECONOCIMIENTO
    ", "F_INI_LIQ", "F_FIN_LIQ", "F_INI_EXP", "F_SOLICITUD", "
    F_VENCIMIENTO", "F_NOTIF") AS
SELECT DISTINCT
DPE.EXPEDIENTE AS EXPEDIENTE,
DPE.OBJETO AS OBJETO,
U.NOMBRECOMPLETO AS NOMBRECLIENTE,
TEXP.DESCRIPCION AS TIPO_EXP,
ESEX.DESCRIPCION AS ESTADO,
DPE.CARPETA AS CARPETA,
TACT.DESCRIPCION AS ACTIVIDAD_PRINCIPAL,
TIO.DESCRIPCION AS IVA_OCUP,
TIA.DESCRIPCION AS IVA_ACT,
PLO.DESCRIPCION AS PERIODO_OCUP,
PLA.DESCRIPCION AS PERIODO_ACT,
DPE.FECOTORGAMIENTO AS F_OTORGAMIENTO,
DPE.RECONOCIMIENTO AS F_RECONOCIMIENTO,
DPE.FECNOTIFICACION AS F_INI_LIQ,
DPE.FECVENCIMIENTO AS F_FIN_LIQ,
DPE.INICEXP AS F_INI_EXP,
DPE.SOMETIMIENTO AS F_SOLICITUD,
DPE.ULTRESOLU AS F_VENCIMIENTO,
DPE.FECNOTIF AS F_NOTIF
FROM DP.EXPEDIENTE DPE
LEFT JOIN DP.TIPOACTIVPRINCIPAL TACT ON(DPE.CODTIPACTIVPRINCIPAL=
    TACT.CODIGO)
JOIN DP_ESTADOEXPEDIENTE ESEX ON(DPE.ESTADO=ESEX.CODIGO)
JOIN DP_TIPOEXPEDIENTE TEXP ON(DPE.TIPOEXPEDIENTE=TEXP.CODIGO)

```

```

JOIN MLUSUARIO U ON(U.CODUSUARIO=DPE.CONCESIONARIO)
LEFT JOIN LQ.TIPO_IMPUESTOGLOBAL TIO ON(TIO.CODIGO = DPE.
TIPOIGICOCUP)
LEFT JOIN LQ.TIPO_IMPUESTOGLOBAL TIA ON(TIA.CODIGO = DPE.
TIPOIGICTACT)
LEFT JOIN DP.PERIODOLIQ PLO ON(PLO.CODIGO = DPE.PERIODOLIQCOCUP)
LEFT JOIN DP.PERIODOLIQ PLA ON(PLA.CODIGO = DPE.PERIODOLIQCCTIV) ;

```

A.1.9. Vista ACTIVIDADES

```

CREATE OR REPLACE FORCE VIEW
"INTEGRADOSCASTELLON"."ACTIVIDADES" ("EXPEDIENTE" , "F_INI_ACT" , "
F_FIN_ACT" , "TIPO_ACT" , "SUBTIPO_ACT" , "DESCRIPCION" , "
OBSERVACIONES" , "IMPFIJOANUAL" , "IMPFIJOVIG" , "INDTRF" , "TRFMIN" ,
"INDEXLIQ" , "MOTIVOEXLIQ" , "GRAVAMENCIFN" , "TIPOUNIDAD" ) AS
SELECT
DPE.EXPEDIENTE AS EXPEDIENTE,
DPCA.FECINICIO AS F_INI_ACT ,
DPCA.FECFIN AS F_FIN_ACT ,
DPTA.DESCRIPCION AS TIPO_ACT,
DPSTA.DESCRIPCION AS SUBTIPO_ACT,
DPCA.DESCRIPCION AS DESCRIPCION,
DPCA.OBSERVACIONES AS OBSERVACIONES,
DPCA.IMPORTEFIJOANUAL AS IMPFIJOANUAL,
DPCA.IMPORTEFIJOVIGENCIA AS IMPFIJOVIG,
DPCA.INDTRAFICOMINIMO AS INDTRF,
DPCA.TRAFICOMINIMOANUAL AS TRFMIN,
DPCA.INDEXENTOLIQ AS INDEXLIQ,
DPCA.MOTIVEXENTOLIQ AS MOTIVOEXLIQ,
DPCA.GRAVAMENCIFRANEG AS GRAVAMENCIFN,
DPUM.DESCRIPCION AS TIPOUNIDAD
FROM DP.CANONACTIV DPCA
JOIN DP.EXPEDIENTE DPE ON(DPCA.CODEXP = DPE.CODIGO)
JOIN DP.UNIDADMAX DPUM ON(DPUM.CODIGO=DPCA.TIPOUNIDADMAXMIN)
JOIN DP.TIPOACTIVIDAD DPTA ON(DPTA.CODIGO=DPCA.TIPOACTIV)
JOIN DP.SUBTIPOACTIVIDAD DPSTA ON(DPSTA.CODIGO=DPCA.
SUBTIPOACTIV)
;

```

A.1.10. Vista OCUPACIONES

```
CREATE OR REPLACE FORCE VIEW
"INTEGRADOSCASTELLON"."OCUPACIONES" ("EXPEDIENTE" , "F_INI_OCUP" , "
    F_FIN_OCUP" , "TIPO_OCUP" , "TIPO_GRAVAMEN" , "DESC_DETALLADA" , "
    BIEN_PUBLICO" ) AS
SELECT
DPE.EXPEDIENTE AS EXPEDIENTE,
DPCO.FECINICIO AS F_INI_OCUP ,
DPCO.FECFIN AS F_FIN_OCUP ,
TOC.DESCRIPCION AS TIPO_OCUP,
TAO.DESCRIPCION AS TIPO_GRAVAMEN,
DPCO.DESCOCTIV AS DESC_DETALLADA,
DPCO.CODBIENPUBLICO AS BIEN_PUBLICO
FROM DP.CANONOCUPACION DPCO
    JOIN DP_EXPEDIENTE DPE ON(DPCO.CODEXP = DPE.CODIGO)
    LEFT JOIN DP_TIPOACTIVPRINCIPAL TACT ON(DPE.CODTIPACTIVPRINCIPAL=
        TACT.CODIGO)
    JOIN DP_TIPOOCUPACION TOC ON (TOC.CODIGO = DPCO.TIPOOCUP)
    JOIN DP_TIPOACTIVOCUPACION TAO ON (TAO.CODIGO = DPCO.TIPOACTIV)
;
```

A.1.11. Vista BIEN_PUBLICO

```
CREATE OR REPLACE FORCE VIEW
"INTEGRADOSCASTELLON"."BIENPUBLICO" ("COD_BP" , "TIPO_BP" , "DESC_BP" ,
    "TOTALM2" , "F_INI_VIG" , "F_FIN_VIG" , "INSTALACION" , "INSTPRCUSO"
    , "ZONA" ) AS
SELECT
DPBP.CODIGO AS COD_BP ,
TBP.DESCRIPCION AS TIPO_BP ,
DPBP.DESCRIPCION AS DESC_BP ,
DPBP.TOTALM2 AS TOTALM2 ,
DPBP.FECINIVIG AS F_INI_VIG ,
DPBP.FECFINVIG AS F_FIN_VIG ,
DPI.NOMBRE AS INSTALACION ,
DPBP.PORCUSOINST AS INSTPRCUSO ,
BPZ.CODZONA AS ZONA
FROM DP_BIENPUBLICO DPBP
    LEFT JOIN DP_TIPOBIENPUBLICO TBP ON(TBP.CODIGO = DPBP.
        CODTIPOBIEN)
```

```

LEFT JOIN DP_BIENPUBLICO_ZONA BPZ ON(BPZ.CODBIEN = DPBP.
CODIGO)
LEFT JOIN DPP_INSTALACION DPI ON(DPI.CODINSTALACION = DPBP.
CODINST)
;

```

A.1.12. Vista CIFRAS_NEGOCIO

```

CREATE OR REPLACE FORCE VIEW
"INTEGRADOSCASTELLON"."CIFRAS_NEGOCIO" ("EXPEDIENTE" , "F_INI_CIFN" ,
"F_FIN_CIFN" ,"CIFRA_NEGOCIO") AS
SELECT
DPE.EXPEDIENTE AS EXPEDIENTE,
CIFN.FECINICIO AS F_INI_CIFN ,
CIFN.FECFIN AS F_FIN_CIFN ,
CIFN.CIFRANEGOCIO
FROM DP_CIFRANEGOCIO CIFN
JOIN DP_EXPEDIENTE DPE ON(CIFN.CODEXP = DPE.CODIGO)
;

```

A.1.13. Vista ESTIBA

```

CREATE OR REPLACE FORCE VIEW "ESTIBA" ("NSOLICITUD" , "ANYO" , "
ESTAESCA" , "ESTADOMA" , "FSALIDA" , "INICIO" , "SALIDA" , "COD_BUQUE"
, "BUQUE" , "GT" , "CALADO" , "MANGA" , "COD_ESTIBADOR" , "
NOMESTIBADOR" , "TONS" , "TEUS") AS
SELECT
TO_NUMBER(NESCALA,99999) NSOLICITUD,
TO_NUMBER (ANYO,9999) ANYO,
ESTADO ESTAESCA,
ESTADOMA,
TO_DATE(FSALIDA) FSALIDA,
INICIO ,
SALIDA,
COD_IMO COD_BUQUE,
BUQUE ,
GT,
CALADO,
MANGA,

```

```

COD_ESTIBADOR      COD_ESTIBADOR,
NOM_ESTIBADOR,
sum(TONS)           TONS,
sum(TEUS)           TEUS
FROM
  ESCALAS, MANIFIESTOS
WHERE
  ESCALAS.EXPEDIENTE = MANIFIESTOS.EXPEDIENTE
  AND TO_DATE(FSALIDA) >= '1-1-2016'
GROUP BY
  NESCALA,
  ANYO,
  ESTADO,
  ESTADOMA,
  FSALIDA,
  INICIO,
  SALIDA,
  COD_IMO,
  BUQUE,
  GT,
  CALADO,
  MANGA,
  COD_ESTIBADOR,
  NOM_ESTIBADOR
ORDER BY ANYO, NESCALA;

```

A.1.14. Vista DETALLE_ESTIBA

```

CREATE OR REPLACE FORCE VIEW "DETALLE_ESTIBA" ("NSOLICITUD" , "ANYO" ,
"ESTAESCA" , "ESTADOMA" , "FSALIDA" , "INICIO" , "SALIDA" , "
COD_MUELLE" , "NOM_MUELLE" , "COD_BUQUE" , "BUQUE" , "GT" , "CALADO" ,
"ESLORA" , "NOM_TIPO_MERCANCIA" , "COD_ESTIBADOR" , "NOM_ESTIBADOR" ,
"COD_OPERACION" , "TONS" , "TEUS" ) AS
SELECT
  TO_NUMBER(NESCALA,99999)      NSOLICITUD,
  TO_NUMBER (ANYO,9999) ANYO,
  ESTADO ESTAESCA,
  ESTADOMA,
  TO_DATE(FSALIDA) FSALIDA,
  INICIO,
  SALIDA,

```

```

COD_MUELLE,
NOM_MUELLE,
COD_IMO          COD_BUQUE,
BUQUE           ,
GT,
CALADO,
ESLORA,
NOM_TIPO_MERCANCIA,
COD_ESTIBADOR   COD_ESTIBADOR,
NOM_ESTIBADOR,
COD_OPERACION,
sum(TONS)        TONS,
sum(TEUS)        TEUS
FROM
    ESCALAS, MANIFIESTOS
WHERE
    ESCALAS.EXPEDIENTE = MANIFIESTOS.EXPEDIENTE
    AND TO_DATE(FSALIDA) >= '1-1-2016'
GROUP BY
    NESCALA,
    ANYO,
    ESTADO,
    ESTADOMA,
    FSALIDA,
    INICIO,
    SALIDA,
    COD_MUELLE,
    NOM_MUELLE,
    COD_IMO,
    BUQUE,
    GT,
    CALADO,
    ESLORA,
    NOM_TIPO_MERCANCIA,
    COD_ESTIBADOR,
    NOM_ESTIBADOR,
    COD_OPERACION
ORDER BY ANYO, NESCALA;

```

A.1.15. Vista APC_FACTURAS

```

CREATE OR REPLACE FORCE VIEW "INTEGRADOSCASTELLON"."APC_FACTURAS"
  ("SERIE", "ANYO", "NUMFAC", "FECHA", "ESTADOFAC", "EST_BORRADOR",
  "F_LIQUIDACION", "IMPBASE", "IMPIVA", "IMPTOTAL", "IVA", "
  TARIFA", "SUBTARIFA", "EXTENSION", "DESCRIP_TASA", "
  EXPEDIENTEDP", "OBJETODP", "F_INI_SER", "F_FIN_SER", "IDCLIENTE",
  "NOMBRE_CLIENTE", "ANYOESC", "NUMESC", "ESCALA") AS

```

```

SELECT

```

```

F.CODSERFACTURA,
F.ANYOFACTURA,
F.NUMFACTURA,
F.FECFACTURA,
F.ESTADOFACTURA,
B.CODESTADO,
B.FECLIQ,
F.IMPBASE,
F.IMPIVA,
F.IMPTOTAL,
TI.DESCRIPCION,
F.IDTARIFA,
F.IDSUBTARIFA,
F.IDEXTENSION,
TF.DESCRIPCION,
B.EXPEDIENTEDP,
B.OBJETODP,
B.FECINISER,
B.FECFINSER,
B.IDCLIENTE,
B.NOMBRECLIENTE,
B.ANYOESC,
B.CODESC,
B.ANYOESC || '/' || LPAD(TO_CHAR(B.CODESC),5,'0')

```

```

FROM LQ_FACTURA F

```

```

JOIN LQ_BORRADOR B

```

```

  ON(F.CODSERFACTURA=B.CODSERFACEMIT AND F.ANYOFACTURA=B.
  ANYOFACEMIT AND F.NUMFACTURA=B.NUMFACEMIT)

```

```

LEFT JOIN LQ_TIPO_IMPUESTO TI

```

```

  ON(B.CODTIPIMPUESTO=TI.CODIGO)

```

```

LEFT JOIN LQ_TARIFA TF

```

```

  ON(B.IDASPECTOTARIFA=TF.IDTARIFA AND B.IDASPECTOSUBTARIFA=TF
  .IDSUBTARIFA AND B.IDASPECTOEXTENSION=TF.EXTENSION)

```

```

ORDER BY 1;

```

A.1.16. Vista FACTURAS

```
CREATE OR REPLACE FORCE VIEW "INTEGRADOSCASTELLON"."FACTURAS" ("
SERIE", "ANYO", "NUMFAC", "FECHA", "ESTADOFAC", "EST_BORRADOR", "
F_LIQUIDACION", "IMPBASE", "IMPIVA", "IMPTOTAL", "IVA", "TARIFA",
"SUBTARIFA", "EXTENSION", "DESCRIP_TASA", "EXPEDIENTEDP", "
OBJETODP", "F_INI_SER", "F_FIN_SER", "NOMBRE.CLIENTE", "ANYOESC",
"NUMESC", "ESCALA") AS
```

```
SELECT
```

```
SERIE,
ANYO,
NUMFAC,
FECHA,
ESTADOFAC,
EST_BORRADOR,
F_LIQUIDACION,
IMPBASE,
IMPIVA,
IMPTOTAL,
IVA,
TARIFA,
SUBTARIFA,
EXTENSION,
DESCRIP_TASA,
EXPEDIENTEDP,
OBJETODP,
F_INI_SER,
F_FIN_SER,
NOMBRE.CLIENTE,
ANYOESC,
NUMESC,
ESCALA
```

```
FROM APC.FACTURAS
```

```
UNION ALL
```

```
SELECT
```

```
SERIE AS SERIE,
ANYO AS ANYO,
```



```

NUMERO.FACTURA AS NUMFAC,
TO.DATE(FECHA.FACTURA, 'DD/MM/YYYY') AS FECHA,
ESTADOFAC AS ESTADOFAC,
'-' AS EST.BORRADOR,
TO.DATE(NULL) AS F.LIQUIDACION,
IMPORTE.NETO AS IMPBASE,
IMPORTE.IVA AS IMPIVA,
IMPORTE.TOTAL AS IMPTOTAL,
'-' AS IVA,
TASA.TARIFA.CABECERA AS TARIFA,
'-' AS SUBTARIFA,
'-' AS EXTENSION,
'-' AS DESCRIP.TASA,
CASE
    WHEN TASA.TARIFA.CABECERA IN( 'BA', 'BO', 'TA', 'TO') THEN
'-' || SUBSTR(EXPEDIENTE,1,4) || '-' || SUBSTR(EXPEDIENTE,6,5)
    ELSE NULL
END AS EXPEDIENTEDP,
'-' AS OBJETODP,
TO.DATE(FECHA.INICIO, 'DD/MM/YYYY') AS F.INI.SER,
TO.DATE(FECHA.FIN, 'DD/MM/YYYY') AS F.FIN.SER,
USUARIO AS NOMBRE.CLIENTE,
0 AS ANYOESC,
0 AS NUMESC,
CASE
    WHEN TASA.TARIFA.CABECERA NOT IN( 'BA', 'BO', 'TA', 'TO') THEN
EXPEDIENTE
    ELSE NULL
END AS ESCALA

FROM INTEGRA.FACTURAS
GROUP BY SERIE, ANYO, NUMERO.FACTURA, FECHA.FACTURA, ESTADOFAC,
IMPORTE.NETO, IMPORTE.IVA, IMPORTE.TOTAL, TASA.TARIFA.CABECERA,
EXPEDIENTE, FECHA.INICIO, FECHA.FIN, USUARIO
ORDER BY 1
;

```

A.1.17. Vista LINEAS_FACTURA

```

CREATE OR REPLACE FORCE VIEW "INTEGRADOSCASTELLON"."LINEAS_FACTURAS"
("SERIE", "ANYO", "NUMFAC", "LINEA_FAC", "CONCEPTO", "IMPORTE")

```

```

AS
SELECT
B.CODSERFACEMIT,
B.ANYOFACEMIT,
B.NUMFACEMIT,
LF.CODLINEA,
LF.INFOLINEA,
LF.IMPORTE
FROM LQ.BORRADORLINEA LF
JOIN LQ.BORRADOR B
ON(LF.CODBORRADOR=B.CODBORRADOR AND LF.ANYO=B.ANYO)
WHERE B.NUMFACEMIT IS NOT NULL
ORDER BY 1,2,3,4;

```

A.2. Código script QlikView

A.2.1. Main

```

SET ThousandSep = '.';
SET DecimalSep = ',';
SET MoneyThousandSep = ' ';
SET MoneyDecimalSep = ',';
SET MoneyFormat = '#.##0,00 €;-#.##0,00 €';
SET TimeFormat = 'h:mm:ss';
SET DateFormat = 'DD/MM/YYYY';
SET TimestampFormat = 'DD/MM/YYYY h:mm:ss [.fff]';
SET MonthNames = 'ene; feb; mar; abr; may; jun; jul; ago; sep; oct; nov; dic';
SET DayNames = 'lun; mar; mié; jue; vie; sáb; dom';

ODBC CONNECT32 TO [Integrados;DBQ=apdes] (XUserId is *****,
XPASSWORD is *****);

Trimestre:
Mapping Load
    rowno() as Mes,
    'Q' & Ceil(rowno()/3) as Trimestre
    autogenerate(12)
;
ActivDP:

```

```

Mapping Load
    DESCRIPCION as CodACTDP,
    TIPO as Tipo
;
SQL select  DESCRIPCION,COALESCE(TIPO,'R') as TIPO from
    DP_TIPOACTIVPRINCIPAL
;
NombreTipo:
Mapping Load * Inline [
Cod,Tipo
R, Resto
IPC, Intgr . PuertoCiudad
ALM, Almacenamiento]
;
Navegacion:
Mapping Load
    CODIGO asCodigo ,
    DESCRIPCION as Descripcion;
SQL SELECT CODIGO, DESCRIPCION
FROM INTEGRADOSCASTELLON.EST_TIPO.NAVEGACION
;
Pais:
Mapping Load
    CODPAIS as Pais ,
    NOMBRE as Nombre
;
SQL SELECT CODPAIS, NOMBRE
FROM INTEGRADOSCASTELLON.M_PAIS
;

Operacion:
Mapping Load * Inline [
Operacion ,Nombre
D, Des
DT, Des . Tr
E, Emb
ET, Emb. Tr
];

EstadoEscala:
Mapping Load
    CODESTADO as Estado ,

```

```

        NOMBRE as Nombre
;
SQL SELECT CODESTADO, NOMBRE FROM INTEGRADOSCASTELLON.ESTADO_ESCALA;

EstadoAtraque:
Mapping Load
        CODIGO as Estado ,
        NOMBRE as Nombre
;
SQL SELECT CODIGO, NOMBRE FROM INTEGRADOSCASTELLON.ESTADO_ESTADIA;

Puerto:
Mapping Load
        CODPUE as Puerto ,
        Capitalize(NOMBRE) as Nombre
;
SQL SELECT CODPAIS||CODPUE AS CODPUE, NOMBRE
FROM INTEGRADOSCASTELLON.MPUERTO
;
Cliente:
Mapping Load
        CODUSUARIO as IdCliente ,
        NIF as NIF
;
SQL SELECT
        CODUSUARIO,
        NUMDOC||LETRANIF as NIF
FROM M_USUARIO;

Eco1:
Mapping LOAD
        Integra2 ,
        Tarifa
FROM
[.\Economico.xlsx]
(ooxml, embedded labels , table is [I2-Economico]);

Eco2:
Mapping LOAD
        Tarifa2 ,
        Descripcion
FROM

```

```

[.\Economico.xlsx]
(ooxml, embedded labels, table is [I2-Economico]);

PYG:
LOAD Codigo ,
      Nivel4 ,
      Nivel3 ,
      Nivel2 ,
      Nivel1
FROM
[.\PYG.xlsx]
(ooxml, embedded labels, table is Hoja1);
PyG4:
Mapping LOAD Codigo ,
         Nivel4
         Resident PYG
        ;
PyG3:
Mapping LOAD Codigo ,
         Nivel3
         Resident PYG
        ;
PyG2:
Mapping LOAD Codigo ,
         Nivel2
         Resident PYG
        ;
PyG1:
Mapping LOAD Codigo ,
         Nivel1
         Resident PYG
        ;
DROP TABLE PYG;

```

A.2.2. Aux

```

Expedientes1:
Load
    "ACTIVIDAD_PRINCIPAL" as ActividadDP ,
    applymap('NombreTipo', applymap('ActivDP', "
        ACTIVIDAD_PRINCIPAL", 'R'), 'Resto') as ActividadAnalitica ,

```

```

CARPETA as Carpeta ,
ESTADO as EstadoExp ,
EXPEDIENTE as Expediente ,
EXPEDIENTE as ID_Exp ,
AÑO_EXP as AñoExpediente ,
"F_FIN_LIQ" as F_FinLiquid ,
"F_INI_EXP" as F_IniExp ,
"F_INI_LIQ" as F_IniLiquid ,
"F_NOTIF" as F_Notificación ,
"F_OTORGAMIENTO" as F_Otorg ,
"F_RECONOCIMIENTO" as F_Recon ,
"F_SOLICITUD" as F_Solicitud ,
"F_VENCIMIENTO" as F_Venc ,
"IVA_ACT" as IVA_Actividad ,
"IVA_OCUP" as IVA_Ocupación ,
NOMBRECLIENTE as NombreCliente ,
OBJETO as Objeto ,
"PERIODO_ACT" as PeriodoActiv ,
"PERIODO_OCUP" as PeriodoOcup ,
"TIPO_EXP" as TipoExp ,
"F_INI_EXP"-"F_SOLICITUD" as TiempoCiclo
;
SQL SELECT "ACTIVIDAD_PRINCIPAL" ,
CARPETA,
SUBSTR(EXPEDIENTE,3,4) AS AÑO_EXP,
ESTADO,
EXPEDIENTE,
"F_FIN_LIQ" ,
"F_INI_EXP" ,
"F_INI_LIQ" ,
"F_NOTIF" ,
"F_OTORGAMIENTO" ,
"F_RECONOCIMIENTO" ,
"F_SOLICITUD" ,
"F_VENCIMIENTO" ,
"IVA_ACT" ,
"IVA_OCUP" ,
NOMBRECLIENTE,
OBJETO,
"PERIODO_ACT" ,
"PERIODO_OCUP" ,

```

```

"TIPO_EXP"
FROM INTEGRADOSCASTELLON.EXPEDIENTES
WHERE UPPER(EXPEDIENTE) LIKE 'C- %';

Left Join(Expedientes1)
FFactura:
Load
    EXPEDIENTEDP as Expediente ,
    FECHA as F_Fact
;
SQL Select f1.expedientedp , f1.fecha from INTEGRADOSCASTELLON.
apc_facturas f1 where f1.fecha =
(SELECT min(f2.fecha) from INTEGRADOSCASTELLON.apc_facturas f2 where
f1.expedientedp=f2.expedientedp)
;

```

A.2.3. Facturas

```

Facturas:
Load
    SERIE &'-'& ANYO &'-'& NUMFAC as Factura ,
    ANYO AS AñoFactura ,
    Month(FECHA) AS MesFactura ,
    applymap(' Trimestre ',Month(FECHA)) as Trimestre ,
//    ANYOESC as AñoEscala ,
    "DESCRIP_TASA" as DescripcionTasa ,
    ESCALA as Escala ,
    "EST.BORRADOR" as EstadoBorrador ,
    ESTADOFAC as EstadoFactura ,
    EXPEDIENTEDP AS Expediente ,
    EXTENSION as Extension ,
    "F_FIN_SER" as F_FinServicio ,
    "F_INI_SER" as F_IniServicio ,
    "FLIQUIDACION" as FechaBorrador ,
    FECHA as FechaFactura ,
    IDCLIENTE as IdCliente ,
    ApplyMap(' Cliente ',IDCLIENTE) as NIF ,
    IMPBASE as ImporteBase ,
    IMPIVA as ImporteIVA ,
    IMPTOTAL as ImporteTotal ,
    IVA,

```

```

"NOMBRE_CLIENTE" as ClienteFactura ,
NUMESC as NumeroEscala ,
NUMFAC as NumeroFactura ,
OBJETODP as ObjetoDP ,
SERIE as Serie ,
SUBTARIFA as Subtarifa ,
TARIFA as Tarifa ,
applymap( 'Eco1' ,ECO) as TarifaEconomico ,
applymap( 'Eco2' ,applymap( 'Eco1' ,ECO)) as Desc.Economico ,
;
SQL SELECT ANYO,
ANYOESC,
"DESCRIP_TASA" ,
ESCALA,
"EST.BORRADOR" ,
ESTADOFAC,
EXPEDIENTEDP,
EXTENSION,
"F_FIN_SER" ,
"F_INI_SER" ,
"F_LIQUIDACION" ,
FECHA,
IMPBASE,
IMPIVA,
IMPTOTAL,
IVA ,
IDCLIENTE,
"NOMBRE_CLIENTE" ,
NUMESC,
NUMFAC,
OBJETODP,
SERIE ,
SUBTARIFA,
TARIFA,
TARIFA || ' . ' || SUBTARIFA || ' . ' || EXTENSION AS ECO
FROM INTEGRADOSCASTELLON.FACTURAS;

```

LineasFactura :

Load

```

SERIE &'-'& ANYO &'-'& NUMFAC as Factura ,
CONCEPTO as Concepto ,

```



```

IMPORTE as Imp,
"LINEA_FAC" as Linea ,
If (IsNull (TIPO_B0) , '-' ,TIPO_B0) as TipoB0 ,
ApplyMap( 'PyG1' ,PYG) as PYG.N1,
ApplyMap( 'PyG2' ,PYG) as PYG.N2,
ApplyMap( 'PyG3' ,PYG) as PYG.N3,
ApplyMap( 'PyG4' ,PYG) as PYG.N4
;
SQL SELECT
LF.SERIE AS SERIE,
LF.NUMFAC AS NUMFAC,
LF.ANYO AS ANYO,
LF.IMPORTE AS IMPORTE,
LF."LINEA_FAC" AS LINEA_FAC,
LF.TIPO_B0 AS TIPO_B0,
(CASE
    WHEN TIPO_B0 IS NOT NULL THEN
        (CASE
            WHEN TIPO_B0='Emb. Recreo ' THEN
                (CASE
                    WHEN CONCEPTO LIKE '%C: 0,28;% ' THEN
                        F.TARIFA || ' ' || F.SUBTARIFA
                        || ' ' || F.EXTENSION || ' Sasemar '
                    WHEN CONCEPTO LIKE '%A: 0,29;% ' THEN
                        F.TARIFA || ' ' || F.SUBTARIFA
                        || ' ' || F.EXTENSION || ' Buques '
                END)
            ELSE F.TARIFA || ' ' || F.SUBTARIFA || ' ' || F.
                EXTENSION || ' ' || LF.TIPO_B0
            END)
        ELSE F.TARIFA || ' ' || F.SUBTARIFA || ' ' || F.EXTENSION
        END) AS PYG,
LF.CONCEPTO AS CONCEPTO

FROM INTEGRADOSCASTELLON.LINEAS_FACTURAS LF JOIN INTEGRADOSCASTELLON
.FACTURAS F ON(LF.NUMFAC=F.NUMFAC AND LF.SERIE=F.SERIE AND LF.
ANYO=F.ANYO) ;

```

A.2.4. Dominio Público

Expedientes :

Load

```
ActividadDP ,  
ActividadAnalitica ,  
Carpeta ,  
EstadoExp ,  
Expediente ,  
ID_Exp ,  
AñoExpediente ,  
F_FinLiquid ,  
F_IniExp ,  
F_IniLiquid ,  
F_Notificación ,  
F_Otorg ,  
F_Recon ,  
F_Solicitud ,  
F_Venc ,  
IVA_Actividad ,  
IVA_Ocupación ,  
NombreCliente ,  
Objeto ,  
PeriodoActiv ,  
PeriodoOcup ,  
TipoExp  
Resident Expedientes1
```

;

Indicadores :

Load Distinct

```
Carpeta as C ,  
Expediente as Exp ,  
F_IniExp as F_FinTramit ,  
applymap('Trimestre',Month(F_IniExp)) as TrimTramit ,  
year(F_IniExp) as AñoTramit ,  
F_Solicitud as F_IniTramit ,  
NombreCliente as NCli ,  
Objeto as Obj ,  
TipoExp as TExp ,  
TiempoCiclo ,  
F_Fact as LiquidFactura ,  
F_Fact - F_IniExp as TiempoFacturacion  
Resident Expedientes1
```

;

```
DROP TABLE Expedientes1;
```

```
Ocupaciones :
```

```
Load
```

```
    "BIEN_PUBLICO" as BienPublico ,  
    "DESC_DETALLADA" as DescripcionOcup ,  
    EXPEDIENTE as Expediente ,  
    "F_FIN_OCUP" as F_FinOcup ,  
    "F_INI_OCUP" as F_IniOcup ,  
    "TIPO_GRAVAMEN" as TipoGravamen ,  
    "TIPO_OCUP" as TipoOcup
```

```
;
```

```
SQL SELECT "BIEN_PUBLICO" ,  
    "DESC_DETALLADA" ,  
    EXPEDIENTE ,  
    "F_FIN_OCUP" ,  
    "F_INI_OCUP" ,  
    "TIPO_GRAVAMEN" ,  
    "TIPO_OCUP"  
FROM INTEGRADOSCASTELLON.OCUPACIONES;
```

```
BienPublico :
```

```
Load
```

```
    "COD_BP" AS "BienPublico" ,  
    "DESC_BP" as DescripcionBP ,  
    "F_FIN_VIG" as F_IniVigencia ,  
    "F_INI_VIG" as F_FinVigencia ,  
    INSTALACION as Instalacion ,  
    INSTPRCUSO as Instalacion %Uso ,  
    "TIPO_BP" as TipoBP ,  
    TOTALM2 as TotalM2 ,  
    ZONA AS ZonaBP
```

```
;
```

```
SQL SELECT "COD_BP" ,  
    "DESC_BP" ,  
    "F_FIN_VIG" ,  
    "F_INI_VIG" ,
```

```

INSTALACION,
INSTPRCUSO,
"TIPO_BP" ,
TOTALM2,
ZONA
FROM INTEGRADOSCASTELLON.BIENPUBLICO;

CifrasNegocio :
Load
    "CIFRA_NEGOCIO" as CifraNegocio ,
    EXPEDIENTE as Expediente ,
    "F_FIN_CIFN" as F_FinCifra ,
    "F_INI_CIFN" as F_IniCifra
;

SQL SELECT "CIFRA_NEGOCIO" ,
    EXPEDIENTE,
    "F_FIN_CIFN" ,
    "F_INI_CIFN"
FROM INTEGRADOSCASTELLON."CIFRAS_NEGOCIO";

Actividades :
Load
    DESCRIPCION as DescripcionActiv ,
    EXPEDIENTE as Expediente ,
    "F_FIN_ACT" as F_FinActiv ,
    "F_INI_ACT" as F_IniActiv ,
    GRAVAMENCIFN as GravamenCifraNegocio ,
    IMPFIJOANUAL as ImporteFijoAnual ,
    IMPFIJOVIG as ImporteFijoVigencia ,
    INDEXLIQ as IndiceExentoLiq ,
    INDTRF as IndiceTraficoMin ,
    MOTIVOEXLIQ as MotivoExentoLiq ,
    OBSERVACIONES AS OservacionesActiv ,
    "SUBTIPO_ACT" as SubtipoActiv ,
    "TIPO_ACT" as TipoActiv ,
    TIPOUNIDAD as TipoUnidad ,
    TRFMIN as TraficoMin
;

```

```

SQL SELECT DESCRIPCION,
EXPEDIENTE,
"F_FIN_ACT" ,
"F_INI_ACT" ,
GRAVAMENCIFN,
IMPFIJOANUAL,
IMPFIJOVIG,
INDEXLIQ,
INDTRF,
MOTIVOEXLIQ,
OBSERVACIONES,
"SUBTIPO_ACT" ,
"TIPO_ACT" ,
TIPOUNIDAD,
TRFMIN
FROM INTEGRADOSCASTELLON.ACTIVIDADES;

```

A.2.5. Operativa Portuaria

```

Escalas :
Load
    ANYO as AñoEscala ,
    BANDERA as Bandera ,
    BUQUE as Buque ,
    BUQUE.ID,
    CALADO as Calado ,
    COD_IMO as CodImo ,
//    COD_PAIS_DESTINO,
//    COD_PAIS_ORIGEN,
//    COD_PUERTO_DESTINO,
//    COD_PUERTO_ORIGEN,
//    COD_TIPOBUQUE,
//    COD_SERVICIO,
    ESLORA as Eslora ,
    applymap('EstadoEscala',ESTADO) as EstadoEscala ,
    EXPEDIENTE AS Escala ,
    EXPEDIENTE as ID_Escala ,
    FINICIO as FechaIni ,
    FSALIDA as FechaSalida ,
    YEAR(SALIDA) AS AñoSalida ,
    Month(SALIDA) AS MesSalida ,

```

```

    applymap('Trimestre',Month(SALIDA)) as TrimestreEscala,
    GT,
    INICIO as Inicio,
    MANGA as Manga,
    MMPP as MercancíasPeligrosas,
    applymap('Navegacion',NAV_ENTRADA) as NavegaciónEntrada,
    applymap('Navegacion',NAV_SALIDA) as NavegaciónSalida,
    NESCALA as NumEscala,
    NOM_PAIS_DESTINO as PaisDestino,
    NOM_PAIS_ORIGEN as PaisOrigen,
    NOM_PUERTO_DESTINO as PuertoDestino,
    NOM_PUERTO_ORIGEN as PuertoOrigen,
    NOM_TIPOBUQUE as TipoBuque,
    NOM_SERVICIO as ServicioRegular,
    If(IsNull(NOM_SERVICIO)=0,'Si','No') as EsServicioRegular,
    SALIDA as Salida
;
SQL SELECT ANYO,
    BANDERA,
    BUQUE,
    BUQUE_ID,
    CALADO,
    COD_IMO,
    COD_PAIS_DESTINO,
    COD_PAIS_ORIGEN,
    COD_PUERTO_DESTINO,
    COD_PUERTO_ORIGEN,
    COD_TIPOBUQUE,
    COD_SERVICIO,
    ESLORA,
    ESTADO,
    EXPEDIENTE,
    FINICIO,
    FSALIDA,
    GT,
    INICIO,
    MANGA,
    DECODE(MMPP,'S','Si','N','No') AS MMPP,
    NAV_ENTRADA,
    NAV_SALIDA,
    NESCALA,
    NOM_PAIS_DESTINO,

```

```

NOM.PAIS_ORIGEN,
NOM.PUERTO_DESTINO,
NOM.PUERTO_ORIGEN,
NOM.TIPOBUQUE,
NOM.SERVICIO,
SALIDA
FROM INTEGRADOSCASTELLON.ESCALAS;

Left Join( Escalas)
Manifiesto :
Load
    EXPEDIENTE as Escala ,
    NUMDECLARACION as Manifiesto
;
SQL SELECT
    ANYO || '/' || LPAD(TO_CHAR(CODESCALA) ,5 , '0 ') as EXPEDIENTE,
    NUMDECLARACION FROM G3_CABECERA;

Atraques :
Load
    CALADO_ENTRADA as CaladoEntrada ,
    CALADO.SALIDA as CaladoSalida ,
//    COD_ATRAQUE,
//    COD_MUELLE,
//    COD_TIPO_ATRAQUE,
//    COD_USUARIO,
    CONSIGNATARIO as ConsignatarioAtraque ,
    ENTRADA as EntradaAtraque ,
    applymap( 'EstadoAtraque' ,ESTAATRA) as EstadoAtraque ,
    FECHA_FIN as F_FinAtraque ,
    FECHA_INICIO as F_IniAtraque ,
    HORA.FIN as HoraFinAtraque ,
    HORA.INICIO as HoraIniAtraque ,
    NOMMUELLE as Muelle ,
    NOMTIPO_ATRAQUE as TipoAtraque ,
    NORAY_FINAL as NorayFinal ,
    NORAY_INICIAL as NorayInicial ,
    OBSERVACIONES as Observaciones ,
    SALIDA AS SalidaAtraque ,
    TIPO_ESTANCIA as TipoEstancia ,
    ZONA as ZonaAtraque ,
    EXPEDIENTE as Escala

```

```

;
SQL SELECT CALADO.ENTRADA,
        CALADO.SALIDA,
        COD.ATRAQUE,
        COD.MUELLE,
        COD.TIPO_ATRAQUE,
        COD.USUARIO,
        CONSIGNATARIO,
        ENTRADA,
        ESTAATRA,
        EXPEDIENTE,
        FECHA.FIN,
        FECHA.INICIO,
        HORA.FIN,
        HORA.INICIO,
        NOM.MUELLE,
        NOM.TIPO_ATRAQUE,
        NORAY.FINAL,
        NORAY.INICIAL,
        OBSERVACIONES,
        SALIDA,
        TIPO_ESTANCIA,
        ZONA
FROM INTEGRADOSCASTELLON.ATRAQUES;

```

Manifiestos :

Load

```

        BULTOS as Bultos ,
//      COD.CONSIGNATARIO,
//      COD.ESTIBADOR,
//      COD.INST_ESPECIAL,
//      COD.MERC,
//      COD.MUELLE AS COD.MUELLE.MANI,
//      COD.SERVICIO,
//      COD.TIPO_MERCANCIA,
EXPEDIENTE AS Escala ,
FSALIDAESCALA as F_SalidaEscala ,
NOM.CONSIGNATARIO as Consignatario ,
NOM.ESTIBADOR as Estibador ,

```



```

NOM.SERVICIO as ServicioRegularManifiesto ,
NOM.INST_ESPECIAL as InstalacionEspecial ,
NOM.MUELLE AS MuelleManifiesto ,
NOM.TIPO_MERCANCIA as TipoMercancia ,
NOM.LUC as UnidadCarga ,
NUM.LUC as NumUC,
applymap( ' Pais ' ,PAIS) as Pais ,
applymap( ' Puerto ' ,PUERTO) as Puerto ,
TEUS,
IF (TEUS>0,'Si' , 'No') AS LlevaTEUS ,
applymap( ' Operacion ' ,TIPO_OPERACION) as TipoOperacion ,
TONS as Tn,
UC as UC
;
SELECT BULTOS,
"COD_CONSIGNATARIO" ,
"COD_ESTIBADOR" ,
"COD_INST_ESPECIAL" ,
"COD_MERC" ,
"COD_MUELLE" ,
"COD_SERVICIO" ,
"COD_TIPO_MERCANCIA" ,
EXPEDIENTE,
FSALIDAESCALA,
"NOM_CONSIGNATARIO" ,
"NOM_ESTIBADOR" ,
"NOM_INST_ESPECIAL" ,
"NOM_MUELLE" ,
"NOM_SERVICIO" ,
"NOM_TIPO_MERCANCIA" ,
"NOM.LUC" ,
"NUM.LUC" ,
PAIS ,
PAIS || PUERTO AS PUERTO,
TEUS,
"TIPO_OPERACION" ,
TONS,
UC
FROM INTEGRADOSCASTELLON.MANIFIESTOS;

```