

Trabajo fin de Máster

Implementación de un sistema GIS
para la gestión de
recogida de datos de cultivo en campo



upna

Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Máster en Sistemas de Información Geográfica y
Teledetección

Director del Trabajo Fin de Máster: Cesar Arriaga Egüés

Director de la empresa: Enrique Labiano Zabalza

Autor: Andoni Otamendi Pazos

Fecha: 09/09/2019

Implementación de un sistema GIS para la
gestión de recogida de datos de cultivo en campo



Agradecimientos

*“A Cesar, Enrique e Iñaki por
su apoyo y atención durante estos meses”*

Resumen

Este proyecto consiste en una implementación de un sistema desarrollado en QGIS, conectado a una base de datos en PostgreSQL, con extensión PostGIS, para la recogida de datos de cultivo en campo. Este trabajo, realizan técnicos de Estadística Agraria del Gobierno de Navarra, para realizar informes acerca de los cultivos presentes.

En este sistema se podrá editar y añadir información a la base de datos desde el campo mediante QGIS. Este proceso se realizará mediante un formulario diseñado a medida en QGIS. Los datos recogidos se podrán publicar en el servidor Geoserver.

La base de datos se renovará cada vez que se disponga de nueva información acerca de los recintos de las parcelas en Navarra. Esta actualización se ejecutará mediante diferentes consultas en lenguaje SQL, que proporcionarán nueva información a las nuevas tablas, eliminarán información antigua y permitirán explotar la base de datos de forma eficaz.

Palabras clave

QGIS, base de datos, PostgreSQL, sistema, estadística, PAC, Gobierno de Navarra

Abstract

This project consists of an implementation of a system developed in QGIS, connected to a PostgreSQL database, with PostGIS extension, for the collection of crop data in the field. This work is carried out by technicians from the Agricultural Statistics Department of the Government of Navarre, in order to produce reports on the crops present.

In this system, information can be edited and added to the database from the field using QGIS. This process will be carried out by means of a form designed to measure in QGIS. The collected data can be published in the Geoserver server.

The database will be renewed each time new information is available about the enclosures of the plots in Navarre. This update will be carried out by means of different queries in SQL language, which will provide new information to the new tables, eliminate old information and allow the database to be used efficiently.

Key words

QGIS, database, PostgreSQL, system, statistics, PAC, Government of Navarre

Índice

| | |
|---|----|
| 1. Introducción | 9 |
| 1.1 Planteamiento del trabajo..... | 9 |
| 1.2 Resultados de aprendizaje..... | 10 |
| 2. Antecedentes | 11 |
| 2.1 Situación anterior del visor-SITUA..... | 11 |
| 2.2 Situación actual..... | 11 |
| 2.2.1 Fuentes de datos..... | 12 |
| 2.2.2 Integración de los datos..... | 14 |
| 2.2.3 Base de datos..... | 15 |
| 2.2.4 Visor..... | 15 |
| 2.3 Otras Comunidades Autónomas..... | 18 |
| 3. Análisis de requisitos del proyecto | 19 |
| 3.1 Funcionales..... | 19 |
| 3.2 No funcionales..... | 20 |
| 3.2.1 Requisitos de carga y tiempo..... | 20 |
| 3.2.2 Requisitos hardware y software..... | 20 |
| 3.3 Análisis de datos de entrada..... | 21 |
| 3.3.1 Problemática..... | 22 |
| 4. Diseño | 25 |
| 4.1 Motivación para la selección de una base de datos espacial..... | 25 |
| 4.2 Modelo de datos..... | 27 |
| 4.2.1 Conceptual (E/R)..... | 28 |
| 4.2.2 Relacional..... | 38 |
| 4.3 Arquitectura de la solución..... | 44 |
| 5. Implementación | 45 |
| 5.1 Estructura física de la BBDD..... | 45 |
| 5.2 Modelo de datos intermedio..... | 46 |
| 5.3 Proceso de carga, verificación e inserción..... | 51 |
| 5.4 Sistema..... | 55 |
| 5.5 Actualización..... | 60 |
| 6. Tecnologías empleadas | 64 |
| 6.1 Programas..... | 64 |

| | |
|---|-----------|
| 6.2 Lenguajes de programación | 66 |
| 7. Manual de usuario | 67 |
| 7.1 Instalación | 67 |
| 7.1.1 Instalación de PostgreSQL + Postgis..... | 67 |
| 7.1.2 Instalación Geoserver | 68 |
| 7.1.3 Instalación QGIS y conexión con PostgreSQL..... | 68 |
| 7.2 Uso..... | 69 |
| 7.2.1 Modificación de la base de datos | 69 |
| 7.2.2 Publicación de datos | 71 |
| 8. Valoraciones y conclusiones | 72 |
| 9. Bibliografía y webgrafía | 74 |
| Anexos..... | 75 |

Índice de figuras

| | |
|--|----|
| 1. Figura: Menú de la aplicación “SITUA” para elegir imagen de fondo..... | 13 |
| 2. Figura: Imagen NDVI de Abril de 2019 | 14 |
| 3. Figura: Visualización con simbología coloreada de los bordes de las parcelas | 16 |
| 4. Figura: Imagen de la información que contiene el archivo DBF | 17 |
| 5. Figura: Formulario de edición de la base de datos | 17 |
| 6. Figura: Proceso del trabajo que realizan los técnicos..... | 18 |
| 7. Figura: Diagrama de requisitos no funcionales del proyecto (elaboración propia) | 20 |
| 8. Figura: Opciones para navegar por el visor..... | 23 |
| 9. Figura: Parte de la información disponible en el visor | 23 |
| 10. Figura: Etapas para la creación de una base de datos | 28 |
| 11. Figura: Diagrama Entidad/Relación | 29 |
| 12. Figura: Relación Visitas posee Fases..... | 30 |
| 13. Figura: Relación Visitas contiene Variedades | 30 |
| 14. Figura: Relación Visitas proporciona Garantes..... | 31 |
| 15. Figura: Relación Visitas tiene Grupos | 31 |
| 16. Figura: Relación Visitas posee Épocas | 31 |
| 17. Figura: Relación PAC posee <i>Visitas</i> | 32 |
| 18. Figura: Relación Visitas contiene Forzados | 32 |
| 19. Figura: Relación Visitas permanece Abandonos..... | 32 |
| 20. Figura: Relación Visitas pertenece a Municipios..... | 33 |
| 21. Figura: Relación Visitas contiene Productos..... | 34 |
| 22. Figura: Relación Visitas posee Usos | 34 |
| 23. Figura: Relación Visitas contiene SIGPAC..... | 35 |
| 24. Figura: Relación Visitas | 36 |
| 25. Figura: Transformación del modelo conceptual al modelo relacional | 39 |
| 26. Figura: Tablas Forzados, Garantes, Forzados, Épocas y Abandonos..... | 40 |
| 27. Figura: Arquitectura de la solución | 44 |
| 28. Figura: Sinopsis de la nueva base de datos | 46 |
| 29. Figura: Proceso de carga a PostgreSQL | 51 |
| 30. Figura: Imagen del menú de capas cargadas | 56 |
| 31. Figura: Construcción de la relación recintos-visitas | 56 |

| | |
|--|----|
| 32. Figura: Configuración de los campos del formulario..... | 57 |
| 33. Figura: Disposición final del formulario | 57 |
| 34. Figura: Visualización del formulario de un recinto en pantalla | 58 |
| 35. Figura: Opciones del formulario para la edición de atributos | 58 |
| 36. Figura: Expresiones de filtrado para automatizar procesos en el formulario | 59 |
| 37. Figura: Elección de la simbología por cada grupo de cultivos | 60 |
| 38. Figura: Icono de PostgreSQL | 64 |
| 39. Figura: Icono de DBeaver | 64 |
| 40. Figura: Icono de QGIS | 65 |
| 41. Figura: Icono de PostGIS | 65 |
| 42. Figura: Icono de Geoserver | 66 |
| 43. Figura: Icono de SQL | 66 |
| 44. Figura: Edición de la Base de Datos mediante la selección del recinto | 69 |
| 45. Figura: División de la pantalla de QGIS | 70 |

Índice de tablas

| | |
|---|----|
| 1. Tabla: División de las tablas de la base de datos..... | 15 |
| 2. Tabla: Dominios de los atributos del modelo | 37 |
| 3. Tabla: Tabla Municipios | 40 |
| 4. Tabla: Tabla Variedades..... | 41 |
| 5. Tabla: Tabla Productos | 41 |
| 6. Tabla: Tabla Usos | 42 |
| 7. Tabla: Tabla Recintos..... | 42 |
| 8. Tabla: Tabla Visitas | 43 |
| 9. Tabla: Diferencia entre “SITUA” y “visitas” | 48 |
| 10. Tabla: Diferencia entre “SigPac” y “recintos” | 49 |
| 11. Tabla: Diferencia entre “Grupos” y “grupos” | 49 |
| 12. Tabla: Diferencia entre “Fases” y “fases” | 50 |
| 13. Tabla: Diferencia entre “tUsos” y “usos” | 50 |

1. Introducción

1.1 Planteamiento del trabajo

En la actualidad la Sección de Estadística del Departamento de Desarrollo Rural, Medio Ambiente y Administración Local (DRMyAL) perteneciente al Gobierno de Navarra (GN), dispone de un GIS a medida, con un visor y unos datos en ficheros cuyo objetivo es facilitar las comprobaciones que se realizan en campo de la información disponible acerca de los cultivos desarrollados a lo largo de la campaña.

Esta aplicación digital que dispone el GN, se denomina “VisorSITUA” y fue creada hace aproximadamente 10-15 años. Con esta herramienta, se realizan las comprobaciones que desde la oficina son complejas a resolver. Esta aplicación se conecta con la base de datos (formato Microsoft Access) con la que se trabaja editando, cambiando y añadiendo cultivos y sus características tras inspeccionarlos.

Las diferentes fuentes de información con las que cuenta el Negociado de estadística agraria como pueden ser la declaración de la Política Agraria Común (**PAC**) o datos recogidos por el Instituto Navarro para la Transferencia e Innovación en el sector Agroalimentario (INTIA) no permiten obtener estadísticas de gran exactitud, sobre todo en áreas de regadío, por lo que los informes realizados por los técnicos no se ajustan demasiado a la realidad del campo en Navarra. Esta situación, junto con la exigencia de conocer los estados fenológicos de los cultivos, hace necesarias las salidas a campo para poder comprobar cultivos realmente presentes en los distintos recintos **SIGPAC**, y su desarrollo en diferentes estaciones a lo largo del año.

Estas revisiones realizadas desde un vehículo conllevan a editar la base de datos original añadiendo cualquier dato de cultivo; para ello, se utiliza el citado visor que, mediante la ayuda de un GPS, se consiguen visualizar las parcelas cercanas a la trayectoria del vehículo. Este visor, permite acceder a un formulario para la edición de la base de datos.

Sin embargo, en la actualidad los técnicos del Departamento necesitan una aplicación, con más funcionalidades y sobre todo con más agilidad que la aplicación disponible, ya que las salidas a campo requieren de muchas revisiones (paradas muy frecuentes para poder realizar comprobaciones de cada recinto) y la herramienta actual ralentiza dicho trabajo (los casi 1 millón de registros en la **base de datos** y la antigüedad de la aplicación afectan al rendimiento en el uso de la aplicación).

Para ello este documento tratará de explicar la definición de un sistema **GIS**, basado en desarrollo abierto y gratuito como es QGIS (añadiendo un formato nuevo de base de datos), que permita pasar de la representación gráfica a la información de los atributos (y viceversa). Este nuevo sistema posibilitará desarrollar unas comprobaciones en campo más ágiles, además de añadir más funcionalidades y conectar de un modo más eficiente con la base de datos para poder modificarla.

Además, se podrá hacer consultas (el nuevo formato de la base de datos estará basado en lenguaje **SQL**) explotando así, la base de datos.

1.2 Resultados de aprendizaje

Este presente trabajo ha integrado numerosos aspectos aprendidos durante el máster en Sistemas de Información Geográfica y Teledetección (MUSIGT). Durante este curso se han adquirido unos conocimientos que muchos de ellos, han sido aplicados en este proyecto. Los contenidos del máster son tan variados que han permitido realizar el trabajo que tenemos entre nuestras manos, ya que cabe mencionar, que si por algo se caracteriza este trabajo es por integrar diferentes programas, lenguajes y aspectos para poder aunarlos y realizar una publicación en completa relación con el máster.

Este trabajo guarda una fuerte relación con la asignatura denominada “Bases de datos Geográficas”, en la que se ha basado este trabajo. El fundamento de este documento se basa en un sistema GIS integrando una base de datos a la que se añade una extensión que permite trabajar con datos geográficos y poder representar gráficamente los datos almacenados.

Se ha conocido y utilizado un modelo conceptual para comenzar a diseñar una Base de Datos (BBDD) y renovar los archivos que se utilizan en la actualidad en el Departamento. Después se ha modelado e implementado una base de datos relacional con elementos geográficos.

En la asignatura mencionada se identificaron diferentes Sistemas de Gestión de Base de Datos que se utilizan en sistemas GIS y Teledetección. Este proyecto los ha empleado para poder almacenar, modificar y extraer información de la base de datos.

Además de todo ello, se han ejecutado diferentes consultas estructuradas para poder verificar y extraer datos requeridos; esta práctica está definida como competencia específica de la asignatura.

Aunque ha estado fuertemente ligado a esta asignatura, este trabajo se ha apoyado en diferentes contenidos de otras materias de MUSIGT. Cabe mencionar, que se ha basado en el sistema QGIS de acceso libre, software trabajado en la asignatura “Sistemas de Información Geográfica”, asignatura que ha permitido desarrollar el sistema obtenido para la captura de datos de cultivo en campo. Se han aplicado técnicas de procesamiento avanzado, trabajando con distintos tipos de datos que puede utilizar un GIS, como por ejemplo, datos vectoriales (polígonos de los recintos de SIGPAC) y ráster (imágenes obtenidas desde el satélite).

Además de la relación con estas dos asignaturas, las diferentes consultas y la edición del software QGIS han sido mediante programación informática. Se ha trabajado mediante lenguajes como SQL para poder explotar la base de datos.

Por último, algunas de las imágenes que los técnicos utilizarán en campo las obtendrán a partir de servicios OGC, de un servidor como GeoServer para poder editar datos geoespaciales desde el campo.

En conclusión, este proyecto se ha compuesto de contenidos, conocimientos y conceptos aprendidos en el máster cursado y gracias a él, se ha podido ejecutar este sistema GIS para poder solucionar un problema en el Departamento del Gobierno de Navarra.

2. Antecedentes

Este apartado está destinado a comentar las diferentes situaciones que en relación con este proyecto se dan en diferentes lugares o se han dado anteriormente y sirven como precedente de este trabajo. Se dividirá en diferentes subapartados: primero se explicará la realidad de cada día antes del visor que utilizan actualmente. Después se procederá a describir el trabajo que se realiza en el Departamento mediante la aplicación que disponen y la que se quiere mejorar mediante este proyecto. Por último, se comparará la situación actual correspondiente al trabajo que se realiza en otras Comunidades Autónomas respecto a Navarra.

2.1 Situación anterior del visor-SITUA

El visor que se emplea actualmente en la sección de Estadística, no ha estado presente desde sus comienzos, todo lo contrario, el trabajo que se realiza mediante esta aplicación es bastante reciente.

Antes de esta situación los técnicos realizaban su trabajo de forma diferente, aparte que la información que recibían para elaborar estadísticas también era distinta. La PAC proporcionaba información a nivel municipal, es decir, no se precisaba a nivel de parcela ninguna información acerca del cultivo o cualquier dato semejante. Por lo tanto, la precisión de los datos que se obtenían era bastante inferior a la actual.

Por otro lado, cabe mencionar que, la superficie de segundos cultivos en Navarra antes de la creación del Canal de Navarra (creación de la Primera Fase finalizada en 2011), era menor por la imposibilidad de proporcionar agua a la tierra. Así que, aunque la información que se obtenía era de menos precisión, la realidad del campo navarro era también distinta y no era tan diferente a la proporcionada, por ejemplo, por la PAC.

La información no se verificaba en campo como se hace en la actualidad, los técnicos se ponían en contacto con los dirigentes de las diferentes cooperativas navarras y obtenían datos de sus cultivos y los estados fenológicos de los mismos.

En conclusión, las estadísticas que se obtenían reflejaban alrededor de un 80% la realidad del campo navarro, cuando en la actualidad se está rondando una precisión del 95-98%.

2.2 Situación actual

La aplicación o visor nuevo que será diseñado al concluir este proyecto, tendrá como objetivo adecuarse a los requisitos expresados por los técnicos del Departamento, mejorando el visor actual.

Para construir la nueva aplicación se vuelve imprescindible analizar el funcionamiento del “visorSITUA” y el trabajo que realizan los técnicos para integrar los datos y herramientas que tienen disponibles. En este apartado se desarrollará una descripción del sistema y funcionamiento actual, para poder identificar así todo el proceso que se realiza antes de redactar los informes pertinentes acerca de las estadísticas agrarias de Navarra.

2.2.1 Fuentes de datos

Los técnicos del Departamento que han solicitado este proyecto tienen como objetivo proporcionar informes estadísticos acerca de los cultivos que se observan a lo largo de toda Navarra; su contenido es bastante diverso (por ejemplo, superficies cultivadas por cada especie y variedad, rendimientos anuales estados fenológicos de desarrollo etc.).

Para realizar dichos informes, los datos provienen de diferentes fuentes.

La fuente principal (la que más volumen de información facilita) es la solicitud anual de ayudas de la PAC. Esta solicitud consiste en una comunicación que los agricultores realizan una vez al año a la Unión Europea, con objetivo de recibir las subvenciones de la PAC. Estas subvenciones se aplicarán por diferentes motivos (ayuda al medio ambiente, prácticas beneficiosas relacionadas con el clima etc.) y tienen como objetivo la sostenibilidad del sector primario.

En dicha solicitud, los agricultores deben enunciar los cultivos que se encontrarán en cada recinto de su parcela en el momento de inspección (en Navarra se suelen producir varias revisiones: las primeras se producen en mayo, pero se suelen alargar hasta incluso agosto para comprobar cultivos de verano).

Toda esta información proporcionada a los técnicos de estadística del Departamento es indispensable para poder realizar su trabajo. Sin embargo, con esta información, las conclusiones que podrían sacar los técnicos no serían del todo fiables, ya que un mismo recinto de cada parcela puede contener a lo largo del año diferentes cultivos: por ejemplo, un recinto puede contener algún tipo de forrajera de invierno hasta mayo y después contener un cultivo de verano o de otoño. En este caso, el agricultor declarará el que esté presente en la época de inspección y esa información llegará hasta los técnicos. Sin embargo, la información acerca del resto de cultivos se perdería por no estar incluida en la relación de cultivos de la solicitud de ayudas.

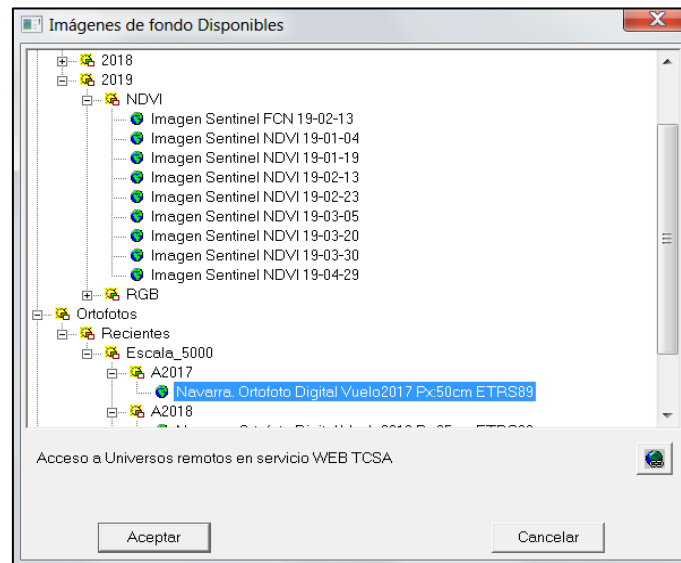
Este problema generalizado a todas las parcelas de Navarra supone una gran pérdida de información estadística, haciendo que las conclusiones e informes redactados dejen de reflejar la realidad de este sector.

La solución, en gran medida, pasa por ir al campo y realizar diferentes revisiones en fechas en las que se puedan encontrar cultivos no reflejados en la solicitud de ayudas PAC. En estas revisiones, los técnicos registran los diferentes cultivos observados en cada recinto, complementando así la información declarada por los agricultores. Por lo tanto, los datos que se recogen en campo también se pueden definir como fuente originaria de los datos.

Otra fuente de datos es el Servicio de Oferta Agroindustrial de la sociedad pública INTIA. Ésta es una sociedad creada por el Gobierno de Navarra y su misión es la transferencia de conocimiento e innovación en el sector agroalimentario que ayude a mejorar tanto la viabilidad como su sostenibilidad, mantener un medio rural vivo respetando el medio ambiente y ofreciendo a la sociedad alimentos y colabora con organismos que *“trabajan en favor del conocimiento y la innovación científico-tecnológica”* relacionadas con el sector agroalimentario. Además de realizar labores de investigación, también asesora a agricultores y empresas, proporcionando

asistencia técnica¹. Esta sociedad proporciona datos de diferentes cultivos pertenecientes a sus socios (socios relacionados con la Comunidad de Regantes) y colectivos de agricultores. Estos datos abarcarán información acerca de unas 50.000-60.000 hectáreas de cultivo.

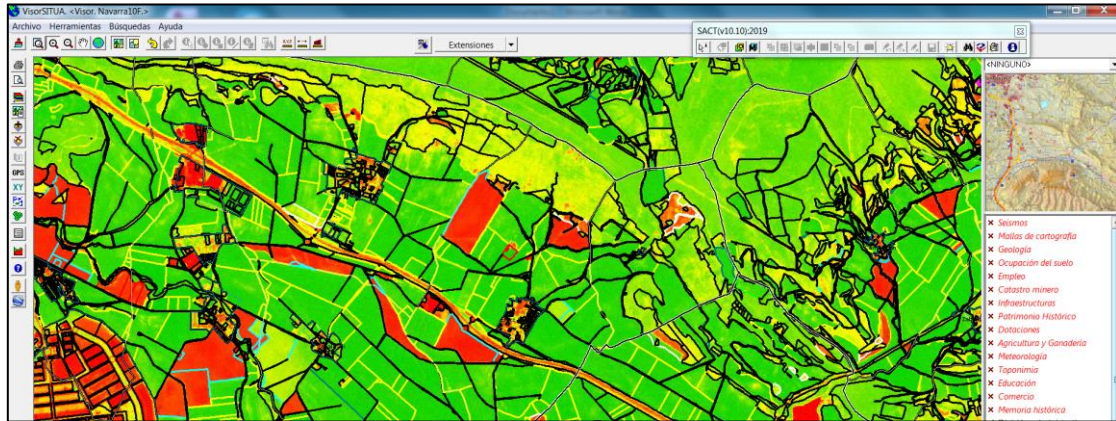
Una fuente adicional de información acerca de los cultivos la construyen las imágenes de satélite. Estas imágenes, obtenidas cada 5 días, permiten hacer un seguimiento desde la misma oficina y sin tener que realizar salidas a campo, sobre el uso de cada recinto y el desarrollo de cada cultivo. Estas imágenes pueden verse en RGB (*Red, Green and Blue*), es decir, como una ortofoto, o pueden calcularse índices de vegetación, como aparece en la siguiente imagen:



1. Figura: Menú de la aplicación “SITUA” para elegir imagen de fondo

Hay que tener en cuenta que Navarra suele aparecer muchos días bajo nubes, por lo que hay bastantes imágenes que no aportan demasiada información, ya que no se pueden apreciar los recintos desde el satélite. Aún así, las imágenes con días despejados proporcionan una valiosa información.

¹ <https://www.intiasa.es/es/que-es-intia.html>



2. Figura: Imagen NDVI de Abril de 2019

Las cooperativas, industrias transformadoras y otras empresas grandes del sector también son fuente de datos. El Departamento de estadística cuenta con bastantes colaboradores (alrededor de 30) que proporcionan información sobre sus cosechas, plantaciones y demás. Estos colaboradores gestionan una zona bastante amplia de suelo cultivable, por lo que el no recoger esta información también supondría bastante pérdida a la hora de analizar las estadísticas.

Por último, hay que mencionar a agricultores individuales que también facilitan datos acerca de rendimientos anuales, hectáreas y demás. Esta información se recoge muchas veces mediante llamadas telefónicas desde la oficina.

Por lo tanto, la base de datos se nutre de estas fuentes de datos expuestas y su labor es gestionar y analizar estos datos para que los informes se asemejen lo más posible a la realidad del campo en Navarra. Cabe destacar, que es complejo recoger todos los datos de la Comunidad Foral, pero mediante los datos obtenidos de las fuentes explicadas se puede destacar que se recoge cerca de un 95% de la información total.

2.2.2 Integración de los datos

Una vez que las diferentes fuentes de datos mencionadas hayan provisionado a los técnicos con todos los datos disponibles, éstos deben de integrarlos para poder crear la base de datos con la que se trabajará.

Los diferentes agentes de las empresas y técnicos de INTIA y PAC, mandan datos en diferentes formatos; estos deben de ser tratados antes de integrarlos en la base de datos.

Los técnicos, hacen un preprocesamiento de datos conformando las diferentes entidades que luego se podrán observar en la base de datos. Una vez que estos datos han sido uniformados (los formatos son muy diferentes), se exportan a la base de datos creada en formato Access. Esta nueva información, se debe cruzar con la información que ya se tiene en la base de datos.

En este caso, hay información permanente en la base de datos y una nueva información adquirida de las fuentes de datos para poder actualizar la ya existente. Este es un proceso bastante arduo para los técnicos, ya que esta actualización se debe realizar manualmente. Si se producen diferencias en información como, por ejemplo, superficie declarada, se deberá analizar qué información se quedará en la base de datos. Esta información que perdurará en la base de datos será la que

proporcione más fiabilidad. Esta fiabilidad, se debe de evaluar haciendo comprobaciones en oficina con el visor y diferentes herramientas.

2.2.3 Base de datos

La base de datos se confeccionará, por tanto, con la información permanente que perdura año tras año y con la nueva información de cada año. Esta base de datos se puede dividir en dos: por un lado, tablas con información auxiliar o de “catálogo”, que aprovisionan a las tablas principales de diferentes códigos; por otro lado, dos tablas principales: una que contiene los datos de SIGPAC (con todos los recintos, y características de Navarra correspondientes a la campaña considerada) y otra tabla donde se integran los datos disponibles acerca de la situación de los cultivos (proporcionados por las distintas fuentes de información) y codificados de acuerdo con las tablas “catálogo” mencionadas.

En la imagen siguiente podemos ver la estructura con los títulos de las tablas mencionadas:

1. Tabla: División de las tablas de la base de datos

| Tablas catálogo | Tablas principales |
|-----------------|--------------------|
| Abandonos | SITUA |
| Fases | SIGPAC |
| Forzados | |
| Garantes | |
| Grupos | |
| Invernaderos | |
| Municipios | |
| Productos | |
| Simplificación | |
| Usos | |
| Variedades | |

Cabe destacar que el nombre de algunas de estas tablas reflejadas en la imagen no refleja fielmente lo que las tablas contienen, por lo que será otro punto, a modificar, en el nuevo modelo de datos.

2.2.4 Visor

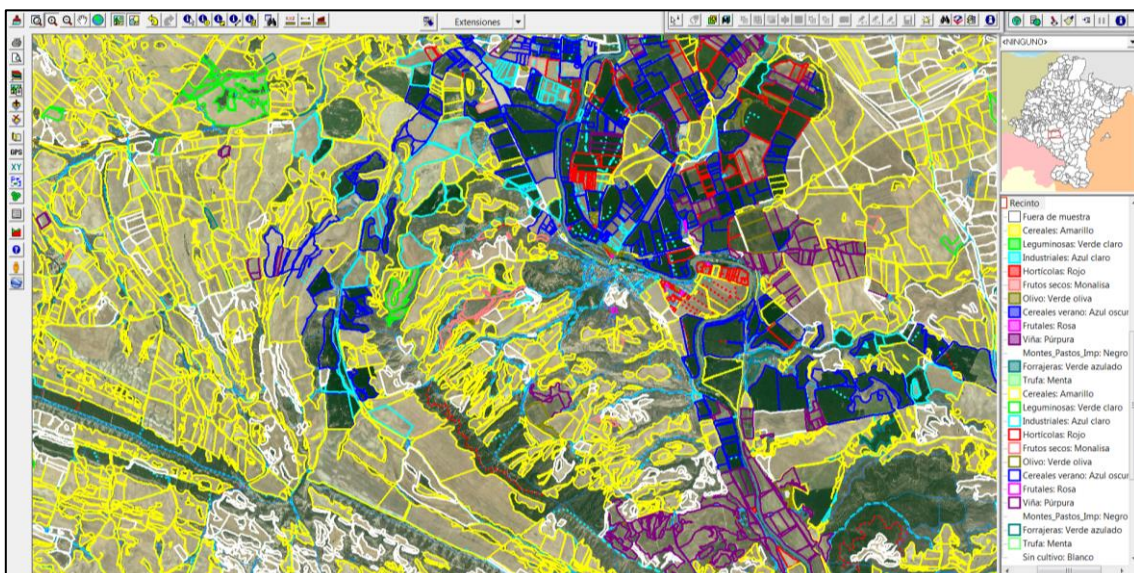
Como se ha explicado en este mismo apartado en la sección de fuentes de datos, los técnicos deben salir al campo para recoger información que no ha sido adquirida por otras fuentes.

Estas salidas se realizan desde la oficina del Departamento en Pamplona hasta cualquier zona de Navarra (mayoritariamente se sale a la Ribera (Zona Sur)), mediante un vehículo todoterreno. Además de este todoterreno, las herramientas que se utilizan suelen ser un ordenador portátil y una antena con conexión para proporcionar señal GPS. Tanto la antena como el ordenador se conectan al vehículo

para proveerse de energía. Son necesarias dos personas: una para conducir el vehículo y otra para registrar información en el ordenador.

Los técnicos desde la oficina ya han cargado los datos a la base de datos, para poder añadir información en una nueva salida a campo. Desde la oficina deciden qué zona comprobar. En esta zona se adentran en caminos de parcelaria, para poder analizar el estado fenológico de los cultivos, los rendimientos potenciales en el momento de la visita y el tipo de cultivo desde cerca.

En este proceso, las parcelas que contienen cereales y ya están registrados en la base de datos no se suelen revisar, ya que estos cultivos son más sencillos de localizar desde la oficina (mediante imágenes de satélite). Por lo tanto, en estas salidas se revisan los demás cultivos, que han podido ser cosechados, recién plantados, estar en proceso de crecimiento o madurez. Los técnicos identifican los cultivos y deben trasvasar esa información a la base de datos. Para facilitar esa identificación, el visor muestra una simbología con el contorno de la parcela coloreado; cada color pertenecerá a un grupo de cultivo diferente, como podemos observar en la imagen:



3. Figura: Visualización con simbología coloreada de los bordes de las parcelas

En el ordenador, el “visorSITUA” mediante la señal de GPS sigue el curso del vehículo a través de los diferentes caminos. Al seleccionar cualquier recinto, el visor proporciona información de la base de datos y del DBF.

Este hecho sucede a consecuencia del encaje que une el archivo DBF y la base de datos; este encaje se realiza mediante los cuatro campos identificativos de cada recinto: municipio, polígono, parcela y recinto. Así, unimos la información de los dos archivos e identificamos el mismo recinto. En conclusión, tras este proceso podemos visualizar en la parte baja del visor la información (tanto del DBF como de la base de datos) de un recinto seleccionado.

La necesidad del archivo DBF surge a razón de varios aspectos: Hay que aclarar que este archivo contiene la misma información que la tabla “SITUA”:

| | A | B | C | D | E | F | G | H | I | J | K |
|----|----|-----------|----------|--------------|---------|-----------|----------|--------------|-------------|--------------|-----------|
| 1 | C | MUNICIPIO | POLIGONO | PARCELA, N.6 | RECINTO | RIEGO, N. | USO, C.2 | SUPSP, N.7,2 | JOIN, N.8,0 | PRODUCTO, N. | FASE, N.3 |
| 2 | 31 | 1 | 1 | 227 | 1 | 0ZU | | 0,26 | 0 | 400000 | 22 |
| 3 | 31 | 1 | 1 | 227 | 2 | 0FO | | 2,40 | 1 | 300114 | 22 |
| 4 | 31 | 1 | 1 | 227 | 3 | 0PR | | 0,83 | 2 | 300300 | 22 |
| 5 | 31 | 1 | 1 | 227 | 4 | 0TA | | 0,18 | 3 | 20600 | 14 |
| 6 | 31 | 1 | 1 | 335 | 1 | 0ZU | | 0,07 | 4 | 400000 | 22 |
| 7 | 31 | 1 | 1 | 335 | 2 | 0PR | | 0,03 | 5 | 300300 | 22 |
| 8 | 31 | 1 | 1 | 360 | 1 | 0ZU | | 0,06 | 6 | 400000 | 22 |
| 9 | 31 | 1 | 1 | 360 | 2 | 0IM | | 0,09 | 7 | 400000 | 22 |
| 10 | 31 | 1 | 1 | 360 | 3 | 0TA | | 0,03 | 8 | 10102 | 13 |
| 11 | 31 | 1 | 1 | 360 | 4 | 0ED | | 0,00 | 9 | 400000 | 22 |
| 12 | 31 | 1 | 1 | 1046 | 1 | 0ZU | | 0,00 | 10 | 400000 | 22 |
| 13 | 31 | 1 | 1 | 1046 | 3 | 0PR | | 1,34 | 11 | 300300 | 22 |
| 14 | 31 | 1 | 1 | 1046 | 4 | 0PR | | 1,08 | 12 | 300300 | 22 |

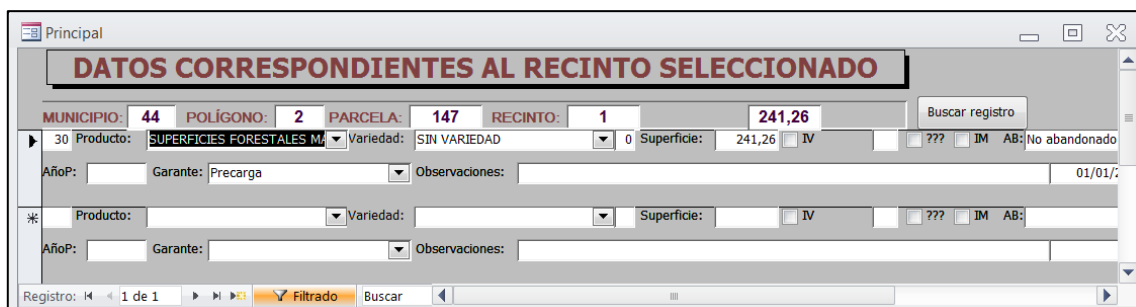
4. Figura: Imagen de la información que contiene el archivo DBF

La razón por la que se trabaja mediante este archivo está relacionada con el formato, ya que el formato *shapefile* necesita un archivo DBF para reflejar esa información en el visor. Por lo tanto, debe de copiarse la información de la tabla principal de *Microsoft Access* a ese formato (.dbf) para poder visualizarla en el visor.

Desde este visor se debe abrir la base de datos una vez seleccionado el recinto donde se haya visualizado un cultivo diferente al que aparece en la base de datos (es decir, un cultivo que por cualquier motivo no se ha facilitado a cualquiera de las fuentes de datos mencionadas anteriormente).

Al abrir la base de datos (mediante un botón de enlace), aparece un formulario con distintos campos a rellenar y con tantos registros como informaciones distintas se tienen para ese recinto. Si no hay información para el recinto considerado, se introducen los datos encontrados: si se ha detectado un error en alguno de los cultivos del recinto, se modifica los datos del registro correspondiente; si se trata de un cultivo de un cultivo adicional, se duplica la información de uno de los registros, y se rellena los campos con los datos del cultivo nuevo.

Normalmente la información que se rellena está relacionada con el tipo de cultivo, el garante (que en este caso es información de campo), la superficie del recinto y alguna observación que se determine oportuna, pero conviene repasar todos los campos propuestos por el formulario para asegurarse que se ha introducido toda la información disponible.



5. Figura: Formulario de edición de la base de datos

Además de añadir información a la base de datos, los técnicos comprueban los estados de los cultivos, información que luego enriquecerá la memoria que realicen.

Por lo tanto, este es el proceso que siguen los datos a lo largo del año comenzando por la integración de las diferentes fuentes y la preparación de la base de datos. Para ello deben actualizar algunos temas manualmente, ya que puede que la información permanente o base, haya sufrido algún cambio. Una vez revisado todos los datos, comienzan a analizar los datos de la base de datos. Este análisis lo realizan contrastando con imágenes de satélite.

Con toda esta información los técnicos proceden a realizar los informes estadísticos; estos informes se realizan de forma semanal, mensual y anual (según el informe correspondiente) y en ellos se detallan las variaciones y novedades en la situación de los cultivos de Navarra (apartado Anexos).



6. Figura: Proceso del trabajo que realizan los técnicos.

2.3 Otras Comunidades Autónomas

La Comunidad Foral de Navarra proporciona información estadística del sector primario navarro al Ministerio de Agricultura y Medio Ambiente, pero, así como Navarra, las demás Comunidades Autónomas realizan el mismo proceso.

Sin embargo, las demás Comunidades no verifican en campo los datos que proporciona la PAC, su información solamente se basa en la declaración de la PAC y en la encuesta sobre Superficies y Rendimientos Cultivos (ESYRCE) que realiza anualmente desde el Ministerio.

Navarra en diferentes aspectos es peculiar, ya que cuenta con sistemas de regadío que hacen posible una agricultura con varias fases a lo largo del año. Sin embargo, en el interior de la península no se encuentran sistemas de regadío y su agricultura es mayoritariamente de secano.

Por lo tanto, podemos definir a Navarra como una de las pioneras en esta realización de comprobaciones a nivel de campo y sobre todo con tanta periodicidad como la realiza. Con ello, los visores y sistemas utilizados para la obtención de datos en campo no son utilizados por otras Comunidades, no para la misma labor, por lo menos.

3. Análisis de requisitos del proyecto

Este siguiente punto corresponderá al análisis de los requisitos que deberá integrar la aplicación objetivo. Requisito, entendido como condición o característica que deberá presentar, junto a sus restricciones, el producto final (entradas, salidas, excepciones, etc.).

Para ello se ha tenido en cuenta diferentes factores: necesidades que los técnicos han precisado y cuestiones interesantes que ayudarán a mejorar la anterior aplicación. Esta información se ha obtenido mediante diferentes reuniones con los técnicos y salidas a campo trabajando con el visor actual; se ha revisado la situación actual del sistema y su organización, así como la documentación existente y los antecedentes de ésta.

Estos requisitos los podemos dividir en dos grupos (por un lado, los requisitos funcionales y, por otro, los requisitos no funcionales) que pasaremos a explicar en las siguientes líneas. Además, se ha insertado un apartado en relación a un análisis que se ha realizado para poder examinar los datos que nos han proporcionado los técnicos, para comenzar a diseñar la base de datos.

3.1 Funcionales

Estos requisitos son entendidos como lo que “debe hacer el sistema”, éste reaccionará a diferentes entradas de usuarios. En este apartado, se detallan los servicios o funciones que proveerá el sistema (por ejemplo, cualquier necesidad mencionada por los técnicos del Departamento que deberá integrar la aplicación).²

Se han recogido los siguientes requisitos funcionales:

- ✓ Visualización del mapa de Navarra dividido en recintos de parcelas
- ✓ Simbología acorde con la del visor actual
- ✓ Disposición de la base de datos en el mismo visor
- ✓ Edición rápida de la base de datos desde un formulario
- ✓ Fácil acceso a la información que se utiliza habitualmente
 - Ortofotos de diferentes fechas
 - Imágenes NDVI
- ✓ Editor 1:N (uno a muchos) para poder introducir diferentes cultivos en un mismo recinto
- ✓ Realizar ampliación/reducción de zoom desde el ratón del equipo
- ✓ Visualización mediante señal GPS
- ✓ Un mantenimiento anual que sea sencillo para la actualización de la información proporcionada por las diferentes fuentes de datos.
- ✓ Posibilidad de dividir el visor en varias ventanas que muestren distintas capas de información y funcionen de forma coordinada.

² <https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales>

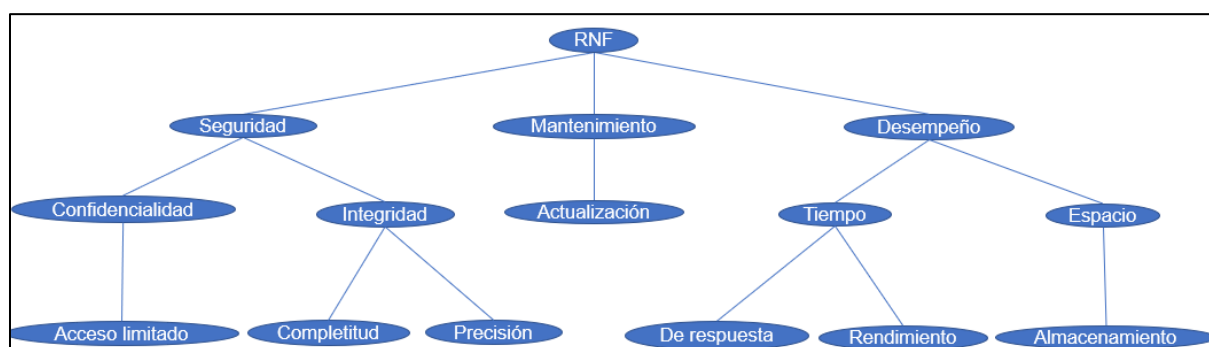
3.2 No funcionales

En cuanto a los requisitos no funcionales, se pueden describir como las condiciones que detallan cómo debe ser el sistema, las restricciones que limitan ciertas elecciones para construir una selección; se pueden mencionar que son atributos relacionados con términos como la “calidad”, “fiabilidad”, “disponibilidad”, “mantenimiento” o “capacidad de almacenamiento”. Por ejemplo, se pueden mencionar como requisitos no funcionales el lenguaje utilizado en este trabajo o la velocidad de procesamiento de los datos en la misma base de datos a la hora de hacer cualquier consulta. Este apartado lo conforman dos subapartados: uno relacionado con el tiempo y carga y otro con las características mínimas que debe de tener el equipo de trabajo.

3.2.1 Requisitos de carga y tiempo

Estos son los requisitos no funcionales de “carga y tiempo” de este proyecto:

- ✓ La aplicación será desarrollada en software libre, como QGIS
- ✓ Navegación eficaz (rendimiento) por el visor, proporcionando funcionalidades como puede ser la ampliación/reducción de zoom
- ✓ Actualización ágil (tiempo de respuesta) de la nueva vista tras el punto anterior
- ✓ Almacenamiento suficiente en la BBDD para cargar todos los registros de Navarra
- ✓ Acceso limitado a la base de datos, dando acceso exclusivamente a los técnicos autorizados



7. Figura: Diagrama de requisitos no funcionales del proyecto (elaboración propia)

3.2.2 Requisitos hardware y software

Este último apartado del análisis de requisitos corresponderá a los requisitos relacionados con el hardware y software. Se van a detallar brevemente los requerimientos mínimos que debería tener un ordenador para poder utilizar sin ningún problema el sistema GIS que se ha diseñado.

Las correspondientes características mínimas del hardware, se recomienda utilizar un procesador superior a 2'5 Ghz, con 4 GB de memoria RAM y 5 GB libres de disco duro como mínimo.

Por otro lado, los requerimientos mínimos correspondientes al sistema operativo base: se recomienda utilizar Windows 7, Windows 8, Windows 10 o Windows vista (32 o 64 bits).

Estas características expuestas son las mínimas para que el sistema soporte el sistema diseñado. A partir de estas características, toda mejora en cada una de ellas conllevará en este caso, a que los procesos que se realizan en la base de datos se procesen a una velocidad superior. Por ejemplo, procesos como la inserción de datos en una tabla o cruces de datos para comprobar la calidad, son procesos que mejorarían con componentes más actuales o con más potencia.

3.3 Análisis de datos de entrada

Este apartado corresponderá al estudio que se ha realizado sobre los datos que nos han proporcionado los técnicos; estos datos y su estructura son los que se utilizan para trabajar.

Los datos están integrados en doce tablas. Si se analiza su estructura, podemos observar que se compone principalmente de una tabla maestra denominada "SIGPAC" añadiendo también al nombre el año al que pertenece dicha información. En esta tabla, nos encontramos con los datos pertenecientes a la Referencia Catastral de cada recinto (compuesto por 4 campos: municipio, polígono, parcela y recinto), su superficie catastral y el uso que ha tenido en esa declaración del año señalado.

Cabe destacar que aparece el campo "JOIN"; este campo les permite a los técnicos poder unir las diferentes tablas (incluso el *shapefile*) por un único código identificador de manera sencilla. Es un campo al que dan bastante importancia.

Aparte de esta tabla "SIGPAC", en este estudio se ha analizado detenidamente la tabla "SITUA". En esta tabla, aparecerán diferentes "apuntes" que los técnicos han realizado tras las diferentes visitas al campo o tras revisar la información desde la oficina mediante imágenes de satélite u otras herramientas.

Los campos en esta tabla son numerosos: incluye, también en este caso, la Referencia Catastral del recinto, su superficie catastral y el uso (al igual que la tabla anteriormente explicada). Además, es evidente la multitud de códigos pertenecientes a las demás tablas "simples", como pueden ser códigos de producto, de variedad, de grupo etc.

Por otro lado, los técnicos ante la abundancia de códigos de cada producto, crean una simplificación para poder agruparlos y trabajar de manera más sencilla. Con ello, añaden columnas de diferentes fases, haciendo referencia a diferentes épocas del año que realizan revisiones.

Tras los códigos, los técnicos añaden ciertos campos, para ampliar la información revisada y facilitar el trabajo, como por ejemplo, los campos denominados "dudoso", "improductivo", "trufa" y "observaciones", para poder escribir cualquier apunte necesario.

Tras este análisis hay muchos datos que serán importantes en nuestro nuevo sistema, pero habrá algunos otros que los obviaremos, ya que no los consideraremos como imprescindibles o porque mediante el nuevo sistema que implementaremos no será necesario integrar dicha información.

Por un lado, cabe destacar que eliminaremos el campo "JOIN". Como antes se ha expresado, es un campo muy importante para los técnicos ya que de manera muy sencilla les permite relacionar las tablas más importantes como son "SITUA" y "SIGPAC" con la capa que visualiza los recintos. En nuestro caso este campo será obviado, ya que el campo que permite visualización será integrado en una tabla de las "maestras" y mediante las reglas de *primary* y *foreign key*, lograremos relacionar sin crear el campo "JOIN".

Por las mismas reglas, se eliminarán también varios campos en los que aparece información duplicada, como puede ser la Referencia Catastral. Hay que tener en cuenta, que los técnicos esta información la necesitaban de esta manera, para poder integrar en el archivo .DBF la información acerca del recinto.

Aparte de estos datos que se eliminarán, toda la información restante será necesaria para nuestro nuevo sistema, aunque como ya se explicará más adelante (en el apartado de implementación, en la sección de modelo intermedio) implementaremos algunos cambios para adaptar la información a las herramientas y al nuevo sistema.

3.3.1 Problemática

El trabajo explicado en las líneas anteriores es el proceso que se realiza para obtener unas estadísticas lo más exactas posibles. Los técnicos de la Sección de Estadística, repiten el proceso todos los años a partir de la carga de los datos procedentes de las distintas de las fuentes.

Este trabajo, sin embargo, se ve obstaculizado por varios problemas que pasaremos a explicar detalladamente. Los técnicos han analizado su proceso de trabajo y han identificado varias mejoras a realizar, relacionadas sobre todo con el visor que utilizan para recoger datos en campo.

Este visor, fue creado hace varios años y sus herramientas y funcionalidades se han quedado anticuadas para las necesidades de los técnicos. Hay que mencionar, que cuando se creó este visor sus funcionalidades y servicios eran de primera calidad, pero el proceso tecnológico llevado a cabo, sobre todo, en la última década ha dejado a esta herramienta obsoleta.

Como ya se ha mencionado, los técnicos recorren caminos recogiendo información de las parcelas que les rodean; estos recintos que se analizan son numerosos en toda Navarra, por lo que cada día que se sale a campo se necesita un mínimo de agilidad en la aplicación o visor para trabajar de forma eficaz.

Éste sería el primer problema planteado y el que podríamos denominar como el problema "esencial". La aplicación carece de agilidad en muchos aspectos del visor. El tiempo de respuesta es bastante alto cuando se intenta navegar por el mapa, es decir, la nueva vista tarda bastante en cargarse. Entre las posibles causas puede estar el mismo visor o los numerosos recintos que debe de cargar en la nueva vista.

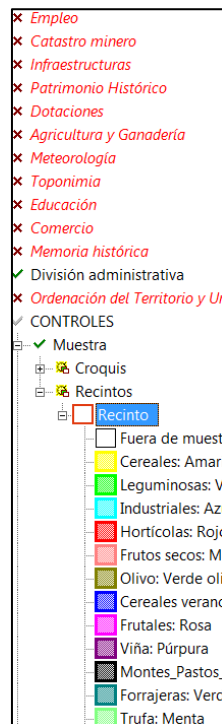
Por otro lado, el visor carece de ciertas funcionalidades. Es verdad, que el visor da la opción de seleccionar botones de zoom para poder ampliar o reducir la vista del

mapa, pero en nuevos visores esta funcionalidad se realiza desde la “rueda” del ratón, funcionalidad que la hace más ágil y sencilla, sin tener que seleccionar ningún botón en el mismo visor. En la siguiente imagen se muestran los botones que el visor facilita para navegar; el seleccionar uno de ellos y aplicar a la vista del mapa, conlleva la ralentización del trabajo.



8. Figura: Opciones para navegar por el visor

Otro problema encontrado por los técnicos se relaciona con la cantidad de información que aparece en el visor. Mucha de esta información es prácticamente dispensable para el trabajo diario en campo. Por ejemplo, el visor contiene en su margen derecha, una cantidad bastante grande de capas disponibles para seleccionar y visualizar. Sin embargo, esta multitud de capas pasan desapercibidas en el trabajo que se realiza. El problema, por otro lado, se centra en que los técnicos deben de desplegar todo el “árbol” de la capa que quieren visualizar, ralentizando el proceso de carga de los mapas.



9. Figura: Parte de la información disponible en el visor

El sistema previamente mencionado del formato *shapefile* también condiciona la carga de datos al visor o aplicación. En este formato se debe utilizar la información que se quiere visualizar; es decir, no se puede adquirir esta información de otra

fuelle, sino que se debe de copiar cualquier información al componente del *shapefile* con extensión “.dbf”.

El DBF sólo admite un registro por recinto, por lo que el formulario de la aplicación no permite registrar más que un cultivo por recinto.

La posibilidad de acceder a un formulario externo es muy lenta, ya que exige abrir la base de datos (Access) cada vez que se quiere acceder al mismo.

Por último, la versión de la aplicación no puede ser renovada (por ello, el objetivo de este proyecto) y se debe de actualizar para poder añadir ciertas funcionalidades. Sobre todo, los técnicos necesitan que sea en un software abierto, como puede ser QGIS, para poder ir actualizando el visor con las nuevas versiones del propio software.

En conclusión, estos puntos de la problemática serán los objetivos que la nueva aplicación deberá superar. Algunos de estos puntos, se resolverán automáticamente solo por el hecho de pasar a QGIS, ya que esta plataforma por defecto integra varias funcionalidades que en el “visorSITUA” no se integran.

4. Diseño

4.1 Motivación para la selección de una base de datos espacial

Este apartado se ha diseñado para poder argumentar el uso de bases de datos espaciales, en vez del formato *shapefile* habitual que se ha estado utilizando últimamente.

Los siguientes argumentos que consideran las ventajas y desventajas de cada formato se han apoyado en un blog de la consultoría ambiental *Geoinnova*³ y en el libro "*Postgis essentials*" (Márquez, 2015)⁴.

El formato *shapefile* ha sido durante muchos años la forma de almacenar y compartir información geográfica. Sin embargo, "*la ausencia de topología, la falta de integridad referencial en los campos y la carencia de mecanismos que aseguren la calidad en la propia información*" ha provocado que las Bases de Datos Espaciales le hayan arrebatado el puesto a los *shapefiles* a la hora de trabajar con un modelo de datos cartográfico.

Los *shapefiles*, "*debido a su simplicidad*" han sido la forma estándar para interactuar con los datos espaciales desde los comienzos del software GIS. Este formato sin embargo tiene una serie de desventajas:

- El *shapefile* se compone de varios archivos, concretamente necesita como mínimo tres: *.shp*, *.shx* y *.dbf*. La primera almacena los objetos geométricos, el segundo se queda con el índice de esos objetos, mientras que en el tercero, se almacenan los atributos de dichas geometrías. Como se ha mencionado, esos tres archivos son la base mínima que compone el formato, pero puede llegar a tener hasta 7 archivos más (*.prj*, *.sbn*, *.ain* etc).
- La segunda desventaja, está ligada con el espacio de almacenamiento, ya que este formato está diseñado para soportar hasta 2GB, por lo tanto, una base de datos más amplia se vería imposibilitada.
- En tercer lugar, no ofrece capacidad para almacenar información topológica, es decir, "relaciones espaciales entre características de diferentes entidades vectoriales conectados en un SIG". Esta desventaja, no permitirá trabajar con algunos tipos de análisis espaciales.
- El *shapefile* no permite un nombre de campo con más de 10 caracteres.
- El quinto punto, se relaciona con las geometrías en las tablas, ya que este formato sólo permite un tipo de geometría por tabla, es decir, no se podrá almacenar una línea y un punto en una misma tabla de atributos.

³ <https://geoinnova.org/blog-territorio/ventajas-de-las-bases-de-datos-espaciales-frente-los-shapefiles/>

⁴ Marquez, A. (2015). *PostGIS essentials*. Birmingham: PACKT publishing.

- Siguiendo con los tipos de atributos los *shapefiles* “no pueden almacenar valores nulos, redondean números, tienen poca compatibilidad con las cadenas de caracteres Unicode. No pueden almacenar fecha y hora en un campo”.

Otra fuente de datos que se ha encontrado en la que se mencionan las restricciones que poseen los *shapefiles*, es la propia web de ESRI (desarrolladora del formato SHP); en ella se puede leer: “Los datos geográficos no son simplemente las entidades y los atributos que puede almacenar un *shapefile*. Por ejemplo, hay anotaciones, relaciones con atributos, relaciones de topología, dominios y subtipos de atributos, precisión y resolución de coordenadas, y numerosas capacidades que son compatibles con las geodatabases, pero no con los *shapefiles*”.

Con esta serie de desventajas (incluso reconocidas por la empresa desarrolladora), queda meridianamente claro que, para gestionar bases de datos activas y de un tamaño amplio el formato SHP conlleva bastantes carencias.

En lugar del formato explicado en las líneas anteriores, se ha apostado por una Base de Datos Espacial, basado en lenguaje SQL:

Antes de explicar las ventajas, comenzaremos por definir este concepto: una Base de Datos Espacial o Geográfica integra datos de cualquier objeto y su localización o coordenadas, es decir, atribuye una representación geométrica en la tierra a esa característica y esta información geográfica se almacena en un nuevo campo, en la tabla que permite representar el objeto, mediante un código binario.

En este proyecto ha optado por este formato, gracias a las ventajas que ofrece, se adecúa perfectamente a las necesidades y requisitos del proyecto (en el apartado Anexos se puede consultar una tabla comparativa entre diferentes formatos). Estas ventajas son:

- Es capaz de realizar la explotación, consulta y cálculos complejos sobre los datos de forma más sencilla. Operaciones complejas e interesantes como los *joins* (relaciones) espaciales, agregaciones, etc, se pueden expresar en una sola línea de SQL; sin embargo, mediante *shapefiles*, ocupa muchas más líneas de código.
- Una de las Base de Datos Geográfica más conocida es PostGIS (extensión de PostgreSQL); la mayoría de los usuarios de esta extensión, instalan sistemas en los que múltiples aplicaciones acceden a los datos, así que tener un método estándar de acceso a SQL simplifica la implementación y el desarrollo.
- Está desarrollada para trabajar con grandes conjuntos de datos; con archivos, pueden ser divididos en varios archivos, pero en una base de datos se puede almacenar todo en una única tabla.
- Permite construir índices con cualquier tipo de datos de forma nativa: “*It builds indexes in almost any kind of data type. Given this flexibility and the fact that the structure of PostgreSQL gives you the chance to build custom functions very tightly to the core, PostGIS could be developed in a natural way*”. Estas construcciones permiten trabajar de forma más eficiente a la hora de explotar la base de datos, ya que permite un acceso más rápido a cada registro de la tabla.

- Las Bases de Datos más utilizadas, contienen numerosas funcionalidades que se pueden llevar a cabo con datos espaciales:
 - Funciones espaciales: para buscar, analizar, convertir y para gestionar datos espaciales.
 - Mediciones espaciales: para cálculos de áreas, longitudes de vectores, distancias entre geometrías etc.
 - Predicados espaciales: respuesta booleana tras la condición expresada en geometrías espaciales
 - Descripción de geometrías: información específica del elemento seleccionado.
- Permite trabajar tanto con información vectorial como ráster.

Marquez (2015) sintetiza todas las ventajas de las Bases de Datos Espaciales en una frase: *“With a GIS database, we can handle geographic data more efficiently because it contains functions and algorithms that make easier to manipulate and analyze it[...]. For the last decade, PostGIS has been used, proved, and improved by a lot of public and private organizations all around the world”*.

En este proyecto, trabajaremos con una Base de Datos bastante amplia, alrededor de 1 millón de registros correspondientes a cada recinto de las parcelas de toda la Comunidad Foral de Navarra. Para su explotación, consulta y edición se ha optado por una Base de Datos Espacial, ya que su almacenamiento, reproducción y el acceso a ella requiere unas características especiales que este formato ofrece.

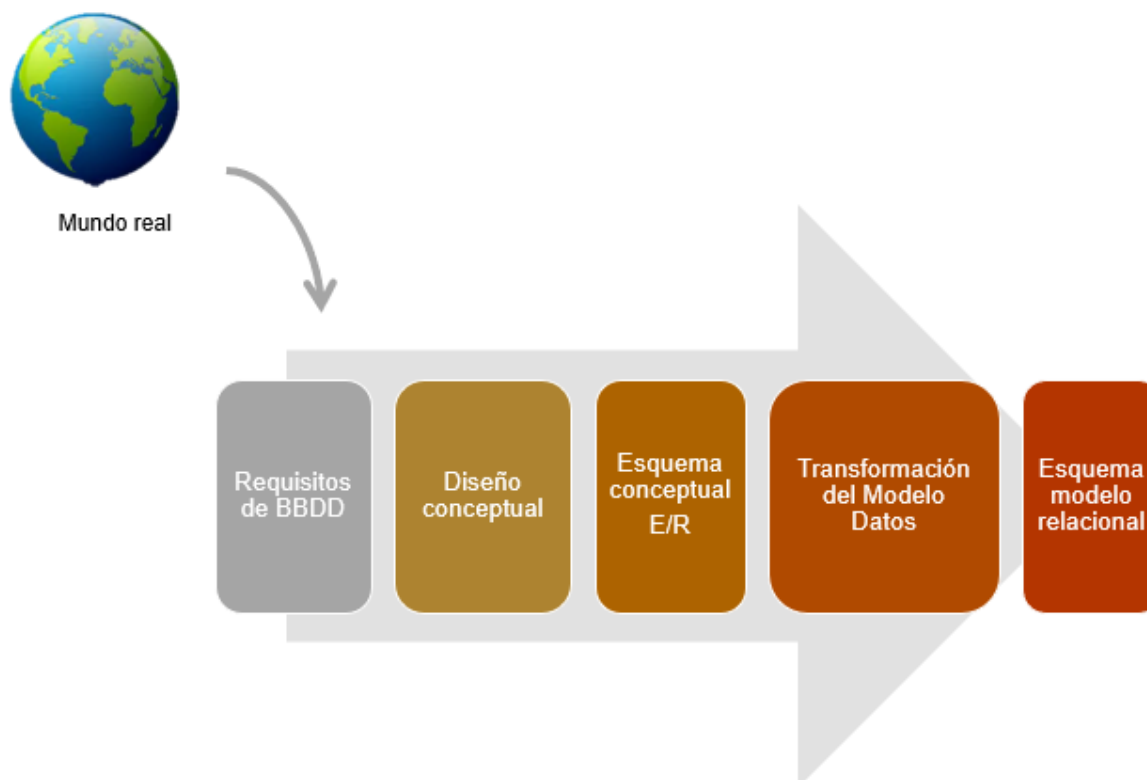
Se ha realizado una pequeña comparativa entre diferentes alternativas para acoger esa base de datos. La cuestión que se nos plantea tiene 3 formatos diferentes de trabajo: por un lado, PostgreSQL + Postgis, en segundo lugar SQLite + Spatialite o por último, la opción de *Geopackage*.

En el apartado “Anexos” se muestra una tabla comparativa con sus limitaciones y potencialidades. Cabe destacar que al final, se ha decantado por PostgreSQL + Postgis, ya que la limitación encontrada (difícil portabilidad) no afecta a este proyecto, ya que todo el sistema estará en un mismo dispositivo portátil. Además, es el formato trabajado en el Máster, por lo que nos ha resultado el formato más familiar de los tres mencionados.

4.2 Modelo de datos

Una base de datos correctamente diseñada es esencial para poder acceder a información de manera eficaz y precisa. Se debe decidir la información requerida, cómo dividirla en diferentes entidades y las diferentes relaciones que puede haber entre éstas.

Los requisitos mínimos que debe cumplir una base de datos están relacionados con la no redundancia de los datos, el acceso autorizado a ella y la relación entre la información de la misma base de datos, entre muchos otros. Establecer estos requisitos es uno de los pasos que se debe de recorrer para diseñar una base de datos, los demás se señalan en esta siguiente ilustración:



10. Figura: Etapas para la creación de una base de datos

En este apartado del proyecto, se explicará detalladamente el diseño técnico que se ha abordado para poder crear una base de datos acorde a las necesidades y cumpliendo ciertos requisitos establecidos por los futuros usuarios, es decir, los técnicos del Departamento.

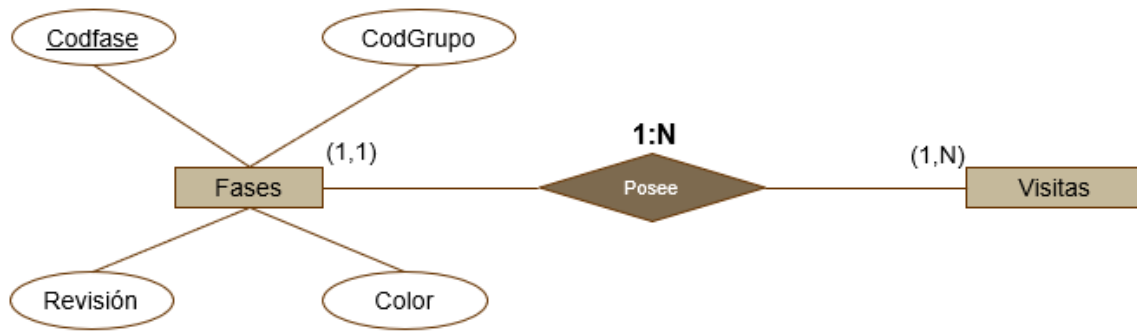
Así, este capítulo se dividirá en dos partes destacadas; por un lado, se precisará el modelo de datos conceptual y por otro lado el modelo de datos relacional.

4.2.1 Conceptual (E/R)

Este tipo de modelo representa “*las entidades importantes y sus relaciones para explorar los conceptos de dominio con interesados en el proyecto que definen el ámbito del problema que tratará la solución del sistema [...]. Consiste en una vista compuesta de información contenida en productos de trabajo de modelado empresarial, requisitos y disciplinas de análisis y diseño que es importante para el desarrollo del modelo de datos*”.⁵

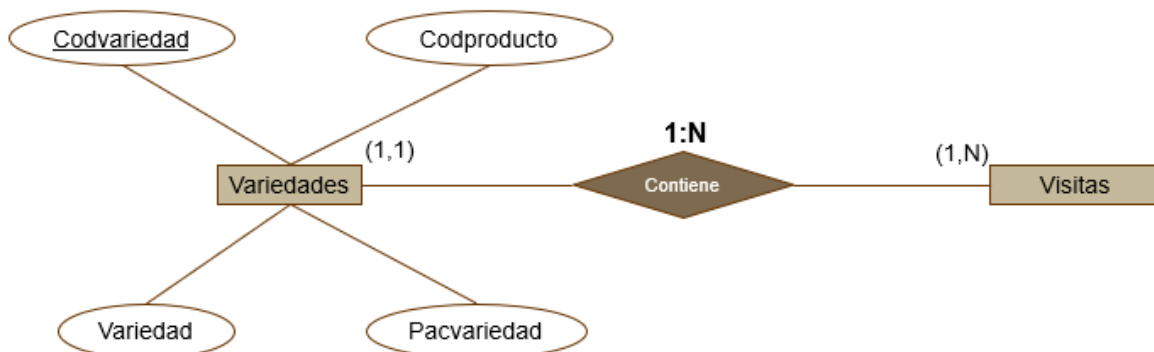
En esta siguiente imagen se refleja el modelo Entidad Relación (E/R) que se ha creado para que cada entidad pueda ser manejada para la obtención de determinados resultados de explotación, gestión, etc.

⁵ Naiburg, Eric J. and Maksimchuk, Robert A. 2001. *UML For Database Design*. New York, NY: Addison-Wesley Publishing Company, Inc.



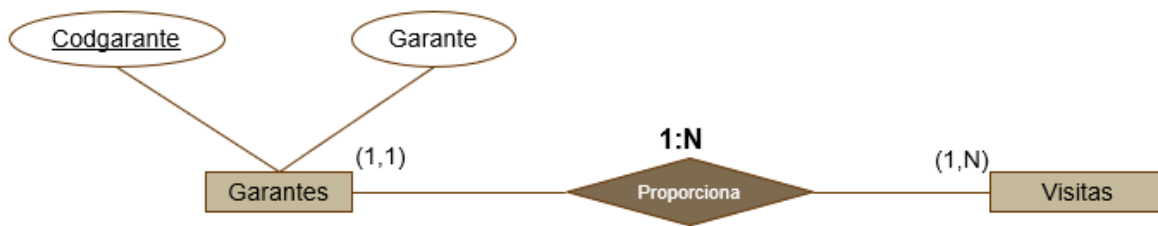
12. Figura: Relación *Visitas* posee *Fases*

La siguiente relación está compuesta por dos entidades, donde la primera de ellas, *Variedades*, se puede denominar como una tabla de “catálogo” o tabla auxiliar de *Visitas*. *Variedades* incluye cuatro atributos que son un identificador único (*Codvariedad*), un código para enlazar con el código la variedad de las ayudas PAC (*Pacvariedad*), con su respectiva información (*Variedad*) y se incluye también un código de producto (*Codproducto*). Cada recinto de *Visitas* contiene una variedad de un producto y el código del respectivo producto. Por ejemplo, un recinto contiene viveros (*Codproducto*) y su variedad pertenece a *Garnacha Blanca* (*Variedad*).



13. Figura: Relación *Visitas* contiene *Variedades*

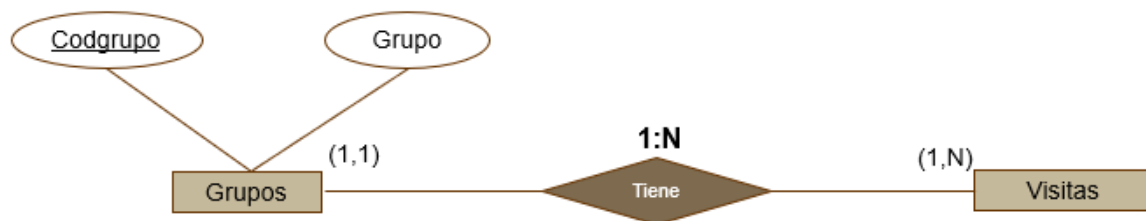
La relación *Visitas-Garantes* está relacionada con las fuentes de datos, ya que los garantes son las fuentes que proporcionarán información acerca de los cultivos. La entidad *Garantes* es una entidad simple que incluye dos atributos: un identificador de cada garante (*Codgarante*) y la observación (nombre) de cada garante. La información de un recinto la puede proporcionar, por ejemplo, el Servicio de Oferta Agroindustrial o la Cooperativa de Ribaforada, entre muchos otros.



14. Figura: Relación *Visit*as proporciona *Garantes*

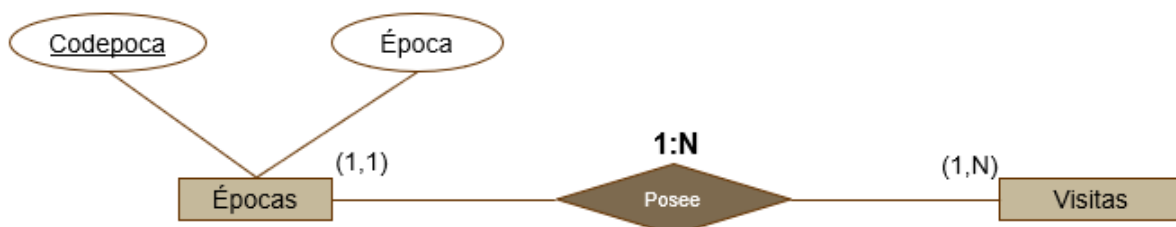
Este siguiente vínculo entre dos entidades corresponde a los diversos *Grupos* que engloban los *Productos*. La tabla *Grupos* tendrá un código identificador (*Codgrupo*) y su respectiva explicación (*Grupo*).

Cada grupo de cultivos, como por ejemplo, *Cereales*, *Leguminosas*, *Industriales* etc. se identificará mediante un código numérico en la relación con *Visit*as. Así, podremos observar cada registro de *Visit*as a qué grupo de cultivos pertenece (por ejemplo: la coliflor pertenece al grupo 7, hortícolas).



15. Figura: Relación *Visit*as tiene *Grupos*

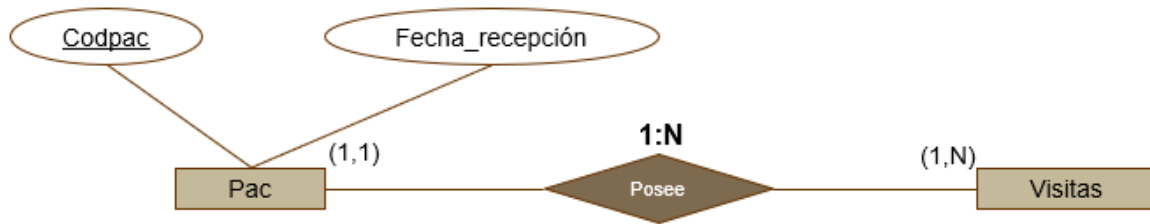
Esta siguiente entidad y relación que se muestra, está relacionada con diferentes *Épocas* del año en que se pueden encontrar los productos de cada registro de *Visit*as. Cada cultivo lo podemos observar en meses concretos normalmente o también podemos encontrar otros cultivos que están presentes permanentemente. Por ello, esta entidad trata de describir cuatro épocas: permanente, primavera, verano y otoño-invierno. Cada época se asignará a un identificador (*Codepoca*).



16. Figura: Relación *Visit*as posee *Épocas*

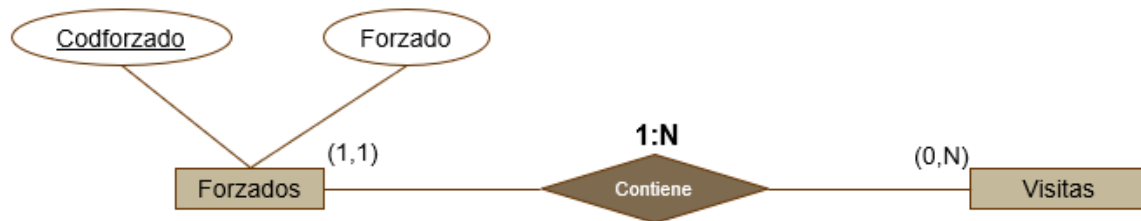
Por otro lado, *Visit*as también posee diferentes códigos de la *PAC*. En esta relación, *PAC* tiene dos atributos donde uno es el código identificativo (*Codpac*) y el otro es la

fecha de recepción de los datos de la declaración. Esta fecha puede ser en julio o en mayo.



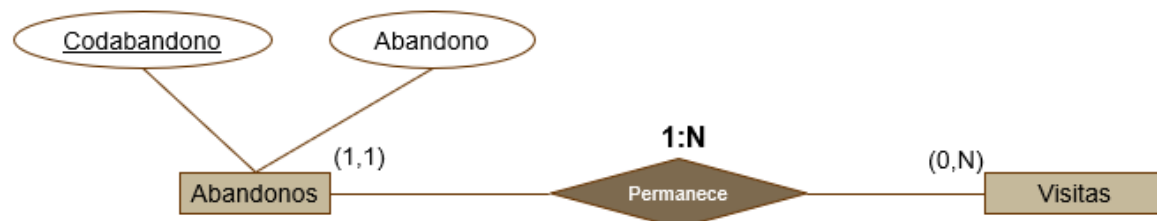
17. Figura: Relación *PAC* posee *Visitas*

Continuando con las entidades auxiliares o de “catálogo” se encuentra también la denominada *Forzados*. Forzado se relaciona con las diferentes metodologías para intensificar la actividad y agilizar el proceso natural del propio cultivo. En esta tabla, se encuentran un identificador de cada tipo de forzado (*Codforzado*) y la explicación del propio (*Forzado*). Entre los diferentes *Forzados* podemos distinguir acolchados, enarenados, túneles, instalaciones fijas o diferentes combinaciones entre las mencionadas.



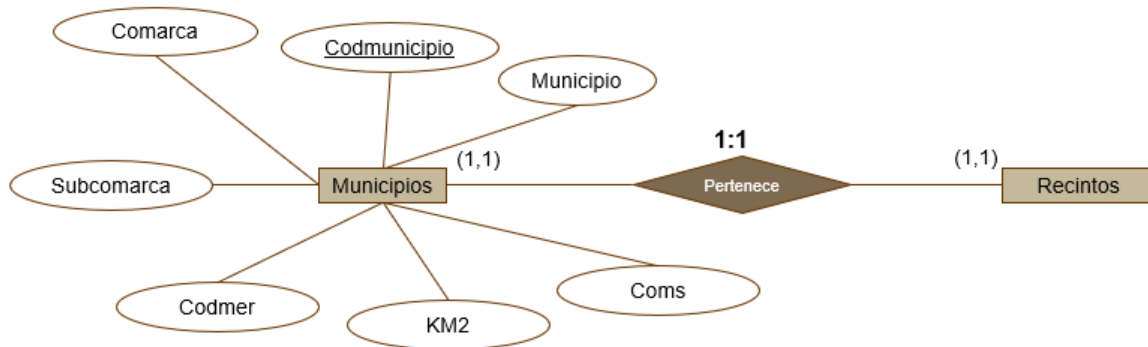
18. Figura: Relación *Visitas* contiene *Forzados*

Como se puede observar las tablas auxiliares predominan en este modelo y la siguiente relación es otro ejemplo de ello. En ella se relaciona cada recinto con diferentes situaciones de abandono en que se encuentran. En la entidad se observa un identificador de cada tipo de abandono (*Codabandono*) y el tipo de abandono (*Abandono*). No son muchas las diferentes situaciones de abandono que se pueden encontrar: puede que no esté abandonado, puede ser de abandono de cosecha, abandono plurianual, cultivo seco o cultivo que ha sido levantado.



19. Figura: Relación *Visitas* permanece *Abandonos*

La entidad *Municipios* congrega todas las entidades municipales de Navarra. Cada recinto en *SIGPAC* pertenecerá a un municipio donde se podrá observar el código identificador (*Codmunicipio*). Además del identificador podremos observar más atributos relacionados con la *Comarca* y *Subcomarca* a la que pertenece cada municipio y la superficie (*Km2*) de éstas.



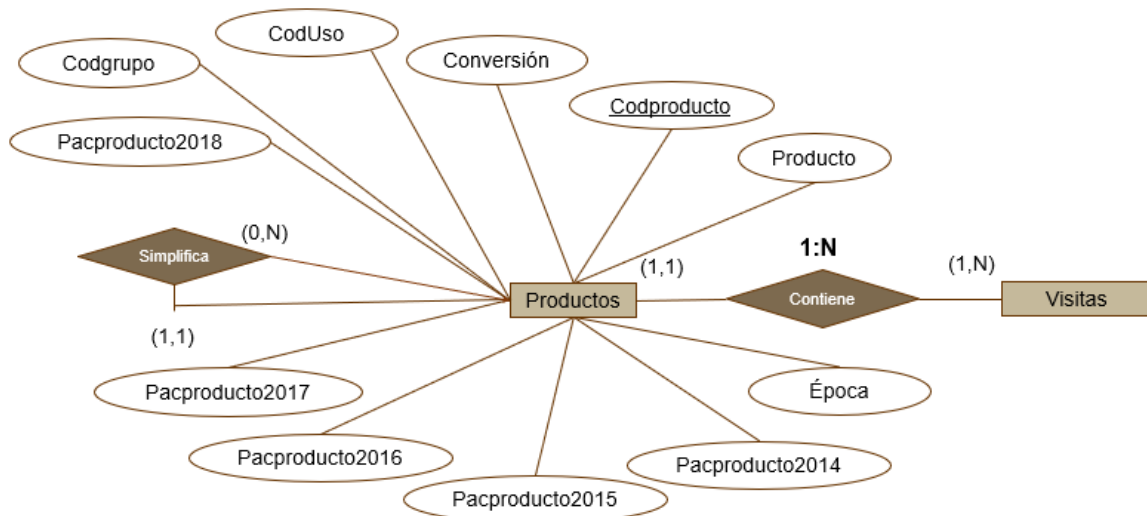
20. Figura: Relación Visitas pertenece a Municipios

Este siguiente vínculo se sale un poco del formato de las tablas auxiliares presentadas hasta ahora; además de contener bastantes más campos, encontramos una relación involutiva.

La entidad *Productos* describe qué cultivos podemos encontrar en cada recinto de *Visitas*. Para ello, encontramos un identificador y su descripción (*Codproducto* y *Producto* respectivamente). Además, diferentes códigos como pueden ser los de uso y grupo (*Coduso* y *Codgrupo*). Por otro lado, *Época* es la estación del año en que se puede encontrar ese producto (aunque hay productos que se pueden encontrar en varias épocas) y los campos más numerosos corresponden a los códigos de los productos en cada año, ya que éstos pueden ir cambiando de año en año.

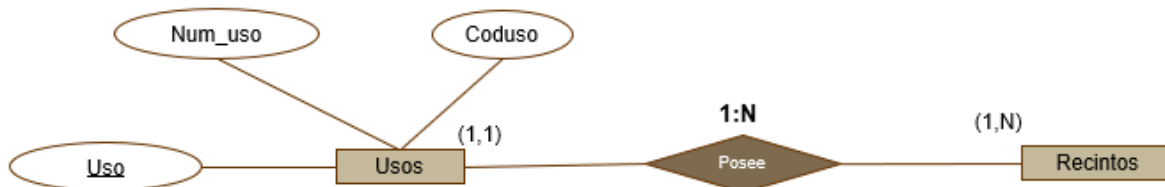
El campo *Conversión*, se crea para evitar problemas técnicos (relacionado con valores nulos del identificador único (Poner valor 999999 al nulo y evitar el campo *Conversión*)).

Por último, pasaremos a explicar la relación involutiva. Esta relación se debe a que hay varios productos que se agrupan (*Codproducto*) creando un nuevo campo identificador. En esta relación, hay productos que se agrupan con otros, pero podemos encontrar a varios productos que no se agrupan en ese nuevo campo.



21. Figura: Relación *Visitas* contiene *Productos*

La próxima relación corresponde a los diferentes *Usos* que pueden poseer los recintos de *SIGPAC*. *Usos* se compone de varios campos donde se encuentran las siglas de cada uno de ellos (*Coduso*) y su respectiva explicación (*Uso*). Además, se añade un código para cada uso/sigla (*CodUso*). Por ejemplo, un recinto cualquiera de Navarra posee un uso FY que corresponderá a frutales, CI que corresponderá a cítricos o TA perteneciendo a tierra arable, entre otros.



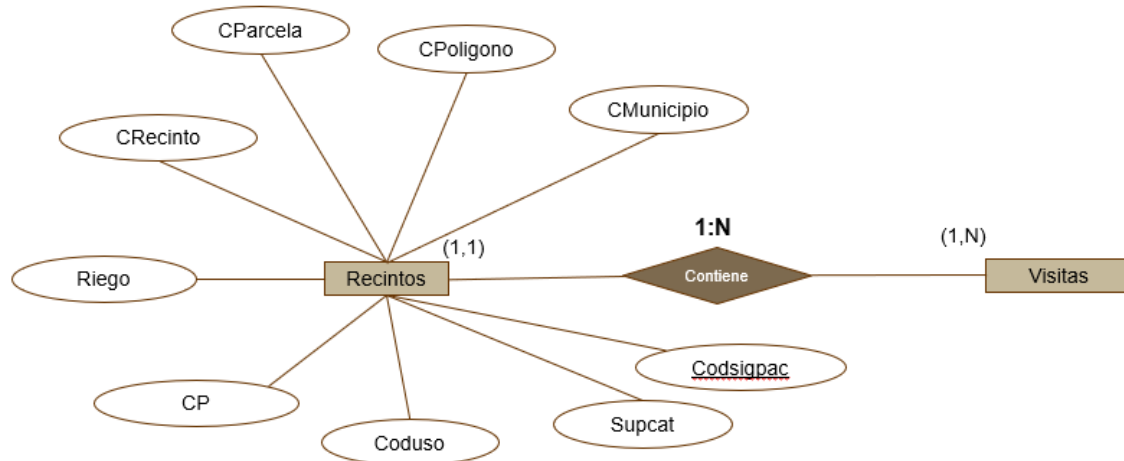
22. Figura: Relación *Visitas* posee *Usos*

La siguiente relación corresponde al vínculo entre las dos entidades principales de este modelo. Cada recinto de *SIGPAC* tendrá su respectivo en *Visitas*, aunque en esta última entidad puede que para cada recinto de *SIGPAC* haya más de un registro, ya que puede obtener varios “apuntes” debido a diferentes *cultivos realizados en el mismo, sea al mismo tiempo o en distintos momentos*, y con ello diferentes *Productos* encontrados a lo largo del año.

SIGPAC se compone de cuatro campos únicos para cada registro, ya que para reseñar cada recinto se enumeran diferentes códigos en los campos *Cmunicipio*, *Cpolígono*, *Cparcela* y *Crecinto*. Para identificar cada registro encontramos el campo *Codsigpac*. También se puede encontrar el campo correspondiente al código de provincia, pero como siempre se trabaja con información de Navarra no es necesario especificar el campo.

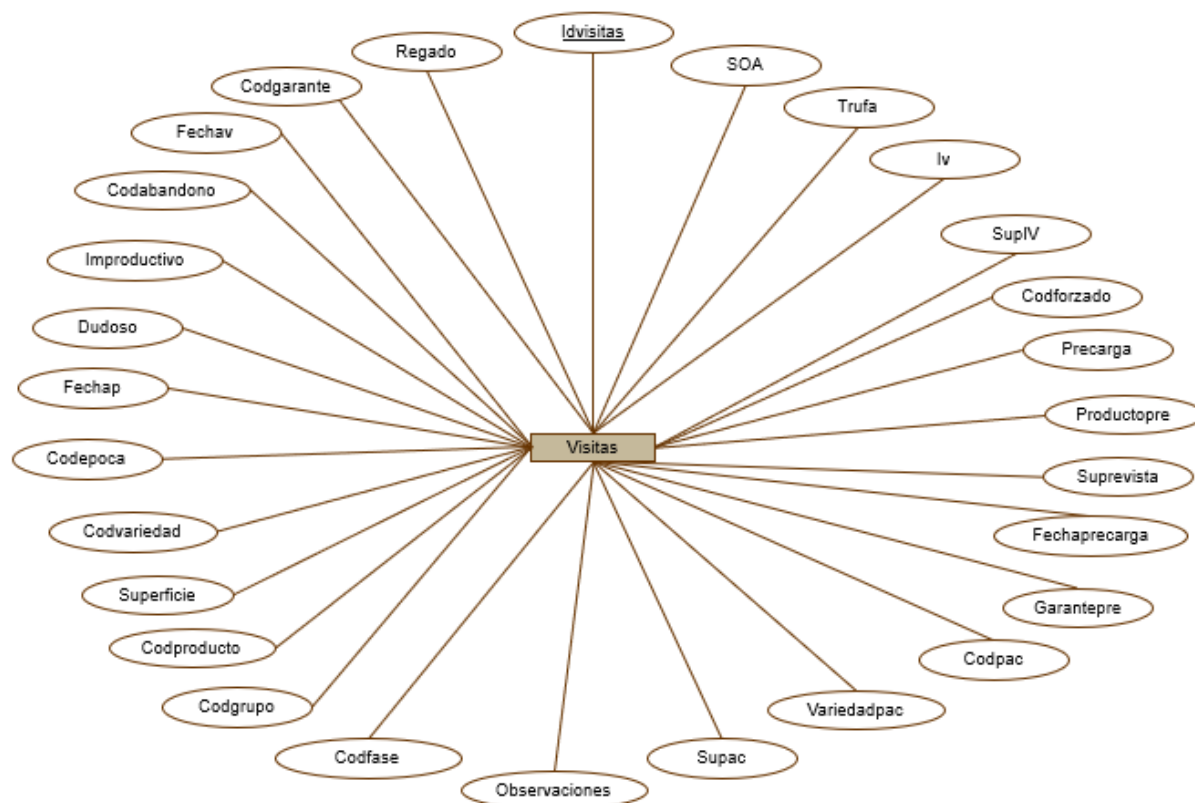
Además de estos campos explicados, se añaden campos como *Riego* donde se determina si ese recinto tiene algún método de irrigación y la superficie de cada

recinto (*Supcat*). También, se puede observar el campo *Coduso* (detallado en la entidad *Uso*) y CP haciendo referencia a la concentración parcelaria, ya que hay recintos que están en proceso de concentración y otro no.



23. Figura: Relación *Visitas* contiene *SIGPAC*

Para finalizar con el apartado Entidad/Relación se detallan los campos que forman parte de la tabla más amplia. Hasta ahora se han descrito las relaciones pertenecientes entre dos tablas, pero esta última no se ha detallado. Esta entidad la componen numerosos campos como se puede apreciar en el siguiente diagrama: entre ellos, destaca el identificador de cada registro de la tabla (*CodVisitas*). A partir de este identificador se añaden diferentes campos relacionados con si el recinto está regado, de que época es, su código de forzado etc.



24. Figura: Relación *Visitas*

Dominios

Los atributos expuestos en cada relación pertenecen a distintos dominios. Éstos tienen ciertas restricciones que pueden ser predefinidas o impuestas por el usuario.

Los atributos del modelo de esta aplicación pertenecen a los siguientes dominios: por un lado, la mayoría de atributos pertenecen al dominio “número entero positivo”: en este dominio encontraremos todos los identificadores y códigos que se utilizan de cada tabla para representar a diferentes descripciones.

Por otro lado, encontramos el dominio “texto” con una longitud máxima de 100 caracteres, donde se compone de las descripciones de cada tabla y campos booleanos. Además, aparece el dominio de “número real” que se compone de varios atributos relacionados la superficie de cada recinto. Por último, encontramos el dominio “Fecha”, donde se compone de atributos relacionados con campos de tipo fecha de plantación, precarga etc. de cada cultivo.

2. Tabla: Dominios de los atributos del modelo

| Número entero positivo | Texto (100) |
|-------------------------------|--------------------|
| Codabandono | Abandono |
| Codfase | Grupo |
| Codforzado | Forzado |
| Codgarante | Garante |
| Codgrupo | Grupo |
| Codproducto | Color |
| Codvariedad | Producto |
| Codepoca | Riego |
| Codmunicipio | Regado |
| Coduso | Variedad |
| Codsigpac | Revisión |
| CodVisitas | Epoca |
| Conversión | Municipio |
| Simplificación | Uso |
| CodGrupo | Soa |
| Turno | Trufa |
| CodProd2018 | Iv |
| CodProd2017 | Precarga |
| CodProd2016 | Dudoso |
| CodProd2015 | Improductivo |
| CodProd2014 | Observaciones |
| CP | Número real |
| CodUSO | Supcat |
| Comarca | Superficie |
| Subcomarca | Km2 |
| Codmer | Supiv |
| Coms | Suprevista |
| Numuso | Supac |
| Garantepre | Fecha |
| Productopac | Fecha_precarga |
| Variedadpac | Fechav |
| Cmunicipio | Fechap |
| Cpoligono | |
| Cparcela | |
| Crecinto | |

4.2.2 Relacional

El segundo apartado de esta sección del modelado de datos corresponde al modelo de datos relacional. Como ya se ha visto anteriormente (figura 10), una vez definidos los requisitos y haber realizado el modelado conceptual, se debe transformar los datos para poder realizar el modelo de datos relacional.

El modelo relacional representa la base de datos como una colección de relaciones y contempla diferentes aspectos en relación a los datos, por ejemplo, la estructura propia de los datos, la manipulación de los mismos y la integridad estableciendo ciertas restricciones que los datos deben cumplir⁶.

Las relaciones se pueden tomar como una “tabla de valores”, donde cada fila de esa misma tabla mostrará ciertos valores relacionados entre sí. Entre sus objetivos se pueden encontrar:

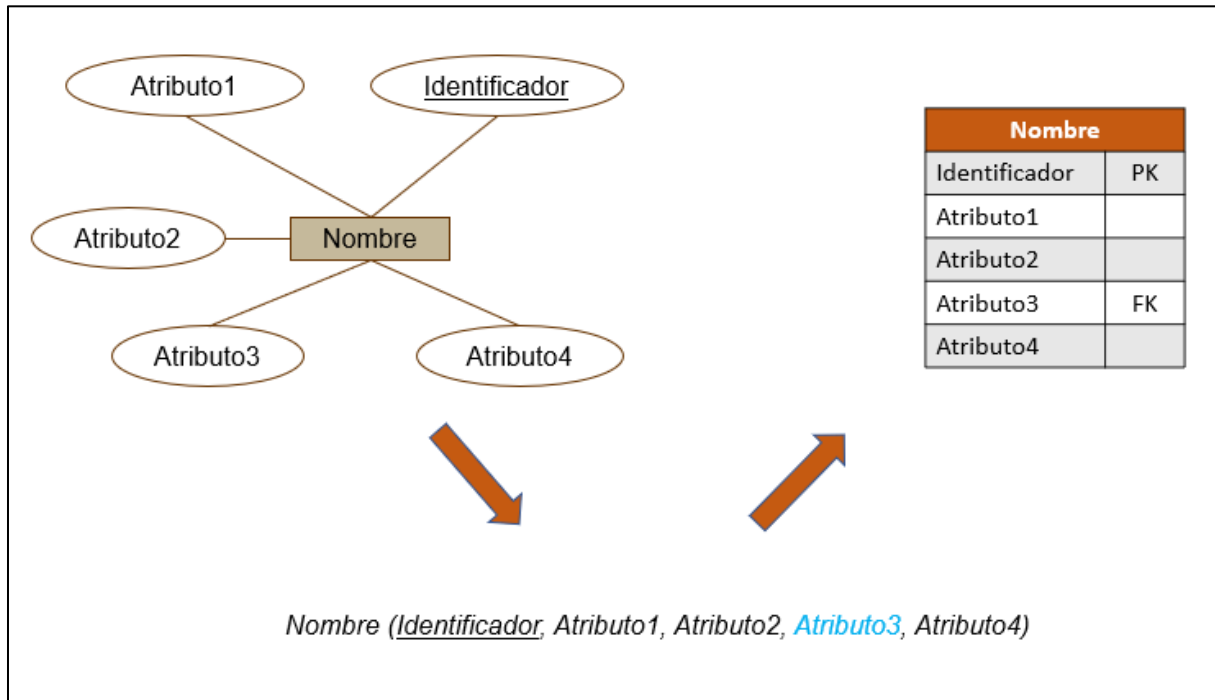
- Favorecer que la estructura de la base de datos sea percibida por el usuario habitual como una estructura lógica que consiste en un conjunto de relaciones.
- Esta estructura debe ser uniforme y simple:
 - La información se representa mediante los atributos de cada relación (tabla).
 - Los valores de los campos se consideran atómicos, es decir, no podrán ser descompuestos dentro de la base de datos.

Con estos objetivos se consigue que cualquier manipulación y explotación de los datos que componen la base de datos, se solucionen de la misma manera.

Cualquier información que contenga la base de datos tiene que poder ser identificada o localizada, por ejemplo, es muy útil el poder localizar cualquier tupla (registro de la tabla). Para ello, en este modelado se utiliza el mecanismo de las claves:

- Clave primaria (PK): será aquella que el diseñador de la base de datos designe para identificar las tuplas de una relación mediante sus valores. Esta clave siempre deberá estar presente en una tabla y podrá ser multicampo (como podremos observar más adelante).
- Clave foránea (FK): clave vinculada a las interrelaciones entre diferentes entidades y almacenará en sus atributos los mismos valores de los atributos que formen la clave primaria de la tupla con la que se relaciona.

⁶ Elmasri, R., & B. Navathe, S. (2010). *Fundamentos de sistemas de bases de datos* (5th ed.) Pearson.



25. Figura: Transformación del modelo conceptual al modelo relacional

Una vez explicada brevemente la parte teórica del modelo relacional, pasaremos a aplicar dichos conceptos en nuestro caso. Así, se detallarán las interrelaciones entre las entidades del modelo realizado. En cada tabla se definirán además de las claves principales (PK), las claves foráneas (FK): campos que se encuentran referenciando a otras tablas. Solo se explicarán las tablas que contengan alguna clave foránea, ya que anteriormente ya se ha comentado cada campo.

En las primeras tablas se exponen las relaciones más simples anteriormente comentadas, donde se encuentran un identificador seguido de una especificación de ese código identificativo.

| Forzados | |
|------------|----|
| Id_forzado | PK |
| Forzado | |

Forzados (Id_forzado, Forzado)

| Abandonos | |
|-------------|----|
| Id_abandono | PK |
| Abandono | |

Abandono (Id_abandono, Abandono)

| Garantes | |
|------------|----|
| Id_garante | PK |
| Garante | |

Garantes (Id_garante, Garante)

| Epocas | |
|----------|----|
| Id_epoca | PK |
| Epoca | |

Epocas (Id_epoca, Epoca)

| Grupos | |
|----------|----|
| Id_grupo | PK |
| Grupo | |

Grupos (Id_grupo, Grupo)

26. Figura: Tablas Forzados, Garantes, Forzados, Épocas y Abandonos

La tabla *Fases* se ha compuesto del campo identificativo con el código de la tabla *Grupos*, además de los campos *Revisión* y *Color* anteriormente mencionados.

3. Tabla: Tabla Fases

| Fases | |
|----------|----|
| Id_fase | PK |
| Codgrupo | FK |
| Revisión | |
| Color | |

Fases (Id_fase, Codgrupo, Revisión, Color)

Esta siguiente tabla, no recibe ninguna clave foránea y muestra como identificador un código asociado a cada municipio.

3. Tabla: Tabla Municipios

| Municipios | |
|--------------|----|
| Id_municipio | PK |
| Municipio | |
| Comarca | |
| Subcomarca | |
| CodMer | |
| Km2 | |
| ComS | |

Municipios (Id_municipio, Municipio, Comarca, Subcomarca, Codmer, Km2, Coms)

En la siguiente tabla se observa una tabla compuesta por un campo identificativo y aparte uno con una clave foránea: *Codproducto*. Este es el código de producto que referencia a la tabla *Productos*, donde como se observa en esa tabla cada producto puede obtener diferentes códigos en los diferentes años.

4. Tabla: Tabla Variedades

| Variedades | |
|-------------|----|
| Id_variedad | PK |
| Codproducto | FK |
| Pacvariedad | |
| Variedad | |

Variedades (Id_variedad, Codproducto, Pacvariedad, Variedad)

La tabla *Productos* muestra varias claves, una primaria y tres foráneas. De estas tres, *Codpgrupo* pertenece a *Grupos*, *Codepoca* pertenece a la tabla *Época* y el código de uso pertenece a la tabla *SIGPAC*. *Codsimplificacion* lógicamente pertenecerá a la tabla *Simplificación*. Los demás campos son originarios de la tabla.

5. Tabla: Tabla Productos

| Productos | |
|-------------------|----|
| Id_producto | PK |
| Producto | |
| Codgrupo | FK |
| Codepoca | FK |
| Coduso | FK |
| Conversión | |
| Codsimplificación | FK |
| Codprod2018 | |
| Codprod2017 | |
| Codprod2016 | |
| Codprod2015 | |
| Codprod2014 | |

Productos (Id_producto, Producto, Codgrupo, Codepoca, Coduso, Conversión, Codsimplificación, Codprod2018, Codprod2017, Codprod2016, Codprod2015, Codprod2014)

Esta siguiente tabla la compone solo una clave principal con otros dos campos originarios de esta misma tabla. No se ha mencionado con las tablas más simples, porque aparte del código identificativo y su explicación (*Uso*) aparece también un número para cada uso.

6. Tabla: Tabla Usos

| Usos | |
|--------|----|
| Id_uso | PK |
| Uso | |
| Numuso | |

Usos (Id_uso, Uso, Numuso)

La siguiente tabla, la denominamos como una de las dos principales del modelo, se puede definir como la tabla “base”. Por ello, no encontramos ninguna clave foránea. Cabe mencionar que la clave primaria en este caso es multicampo, ya que la conforman 4 campos identificativos para cada recinto.

7. Tabla: Tabla Recintos

| Recintos | |
|------------|----|
| Id_sigpac | PK |
| Cmunicipio | |
| Cpoligono | |
| Cparcela | |
| Crecinto | |
| Riego | |
| Coduso | |
| Supcat | |
| CP | |

Sigpac (Id_sigpac, Cmunicipio, Cpoligono, Cparcela, Crecinto, Riego, Coduso, Supcat, CP)

Por último, encontramos la tabla más extensa, es la tabla que se utiliza para la aplicación final. En ella, encontramos un campo identificador y varias claves foráneas provenientes de las diferentes tablas anteriores. Además de ellas, hay campo que los técnicos añaden para recoger información, como por ejemplo si está improductivo, si es dudosa su definición o si se utiliza para la actividad de la trufa.

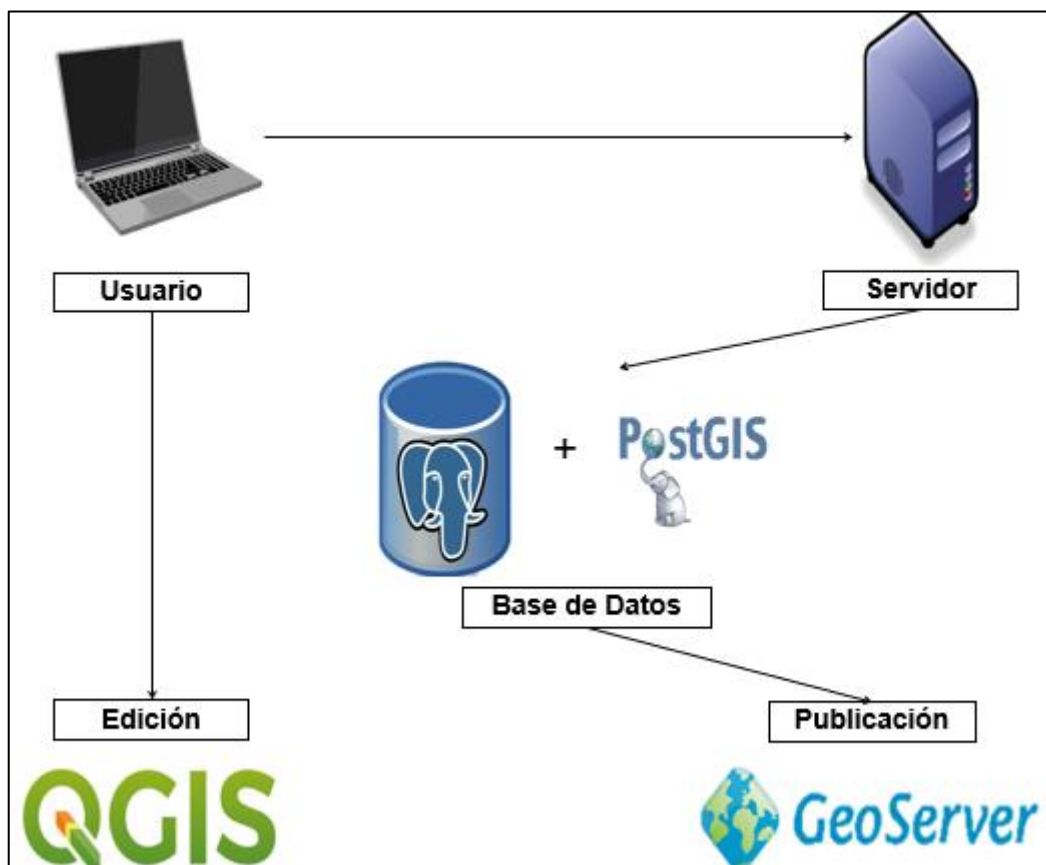
8. Tabla: Tabla Visitas

| Visitas | |
|----------------|----|
| Idvisitas | PK |
| Idsigpac | |
| Regado | |
| Soa | |
| Trufa | |
| Iv | |
| Supiv | |
| Codforzado | FK |
| Precarga | |
| Productopre | |
| Suprevista | |
| Fecha_precarga | |
| Garantepre | |
| Codpac | |
| Productopac | |
| Variedadpac | |
| Supac | |
| Codgrupo | FK |
| Codproducto | FK |
| Codvariedad | FK |
| Codepoca | FK |
| Superficie | |
| Codgarante | FK |
| Codfase | FK |
| Fechav | |
| Codabandono | FK |
| Dudoso | |
| Improductivo | |
| Fechap | |
| Observaciones | |

Ejercicio (*Idvisitas*, *Idsigpac*, *Regado*, *Soa*, *Trufa*, *Iv*, *Supiv*, *Codforzado*, *Precarga*, *Productopre*, *Suprevista*, *Fecha_precarga*, *Garantepre*, *Codpac*, *Productopac*, *Variedadpac*, *Supac*, *Codgrupo*, *Codproducto*, *Codvariedad*, *Codepoca*, *Superficie*, *Codgarante*, *Codfase*, *Fechav*, *Codabanono*, *Dudoso*, *Improductivo*, *Fecha*, *Observaciones*)

4.3 Arquitectura de la solución

La arquitectura del software implica la definición de la solución estructurada que pueda cumplir los requisitos anteriormente expuestos, y además, contribuir a que atributos como rendimiento y seguridad adquieran una calidad lo más eficiente posible.



27. Figura: Arquitectura de la solución

5. Implementación

Tras haber creado el modelo de datos, el siguiente paso será implementarlo en un nuevo sistema. En este apartado, se detallará el proceso que se ha llevado a cabo para poder cargar la nueva base de datos, el modelo intermedio definiendo ciertos cambios entre el modelo relacional y el intermedio y la implementación del nuevo sistema, determinando los procesos de carga y actualización que los técnicos realizan tras recibir nueva información.

5.1 Estructura física de la BBDD

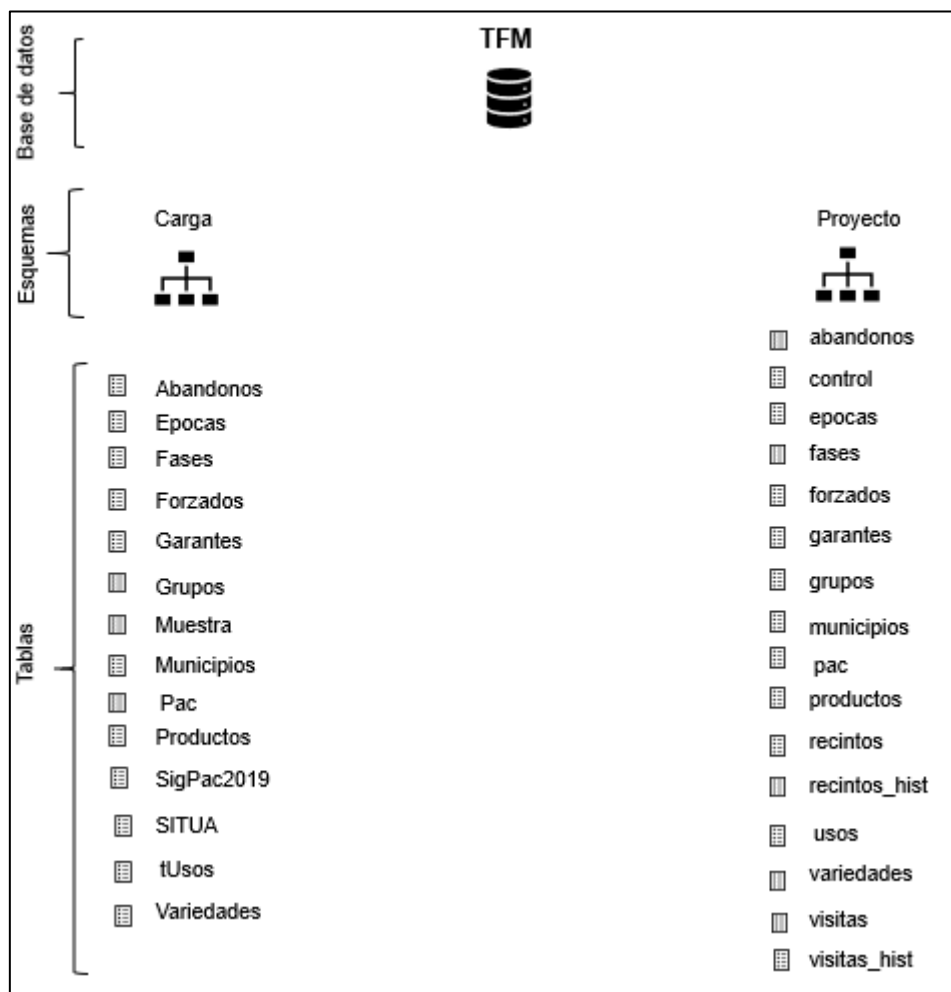
El primer apartado de este sexto punto consistirá en detallar la base de datos que se ha creado y desarrollado para el nuevo sistema que se implementará.

Esta nueva base de datos se ha denominado TFM, en referencia al proyecto que se realiza que es un Trabajo de Fin de Máster. Esta base de datos contiene diferentes esquemas, que los podríamos definir como una “agrupación de tablas u otros objetos”. El hecho de tener diferentes esquemas se debe a que se realizarán diferentes procesos en cada uno de ellos.

El primer esquema denominado “carga”, agrupará las diferentes tablas que se han exportado desde el Access original en el que trabajan los técnicos. Así, la información que contendrá el esquema será el mismo que el original, ya que el trasvase a la nueva base de datos no cambia ninguna información. Por ello, encontraremos los mismos nombres de columnas, el mismo número de columnas y la misma estructura que en el Access. Por otro lado, también se cargará el *shapefile* que se convertirá en una tabla (“muestra”) en este nuevo esquema. En total, encontramos 14 tablas dentro del esquema “carga”.

Cabe destacar que este esquema contendrá estas todas tablas solamente en la carga inicial, ya que una vez realizada ésta, la función de este esquema se basará en contener únicamente la nueva declaración y el nuevo *shapefile* de cada año, para después trasvasarlo al esquema “proyecto”.

Por otro lado, encontramos un segundo esquema denominado “proyecto”. Este esquema es contra el que se trabajará una vez implementado el sistema. Por ello, agrupará las tablas definitivas, es decir, encontraremos algunas de las tablas del anterior esquema, además de algunas nuevas que se crearán. Algunas de las tablas que se trasvasen de un esquema a otro también habrán sufrido diferentes cambios, siempre adecuando a las necesidades de la herramienta. Por otro lado, también habrá cierta información que se trasvasará de forma idéntica, ya que el formato es adecuado y la información es importante para el nuevo sistema. Todos los cambios efectuados entre la carga bruta que se ha realizado al esquema “carga” y el trasvase al esquema “proyecto”, se explicarán en el apartado “Modelo intermedio” de este quinto punto.



28. Figura: Sinopsis de la nueva base de datos

5.2 Modelo de datos intermedio

En el proceso de pasar de un esquema a otro, es decir, del esquema denominado “carga” al esquema “proyecto” se ha creado un modelo de datos intermedio donde se han efectuado los cambios pertinentes para una adecuación a las necesidades de la herramienta y a los requisitos de los técnicos. Por lo tanto, en este apartado se detallará cada cambio que se ha realizado entre los dos esquemas. En general, se han cambiado nomenclaturas, se han creado algunas tablas y se han modificado los tipos de columna. Pasaremos a detallarlos en los siguientes párrafos.

El cambio más sustancial que se ha realizado de un esquema a otro ha sido la integración de la tabla que se ha creado al importar el *shapefile* en la tabla “recintos”. Como ya se ha explicado anteriormente, se ha apostado por una base de datos espacial, por lo que el componente que nos permitirá visualizar la información se encontrará en la tabla mencionada, borrando la tabla original (“muestra”) como se puede comprobar en la figura 27. Esta integración se ha llevado a cabo mediante la referencia catastral que aparece tanto en la tabla “recintos”, como en la tabla “muestra”. Este proceso se realizará varias veces (cada ejercicio se renovará el archivo SHP) como lo podremos comprobar en el siguiente apartado.

La mayoría de cambios que se han realizado han sido para dar un formato unificado a la nomenclatura de las tablas y sus respectivas columnas. Varias veces nos hemos encontrado con un nombre poco identificativo de la información a la que hace referencia en el modelo antiguo. Además de ello, los diferentes nombres empleados en diferentes tablas (claves primarias y foráneas) siendo la misma información hace del modelo antiguo, un modelo poco intuitivo.

Por ello se han aplicado una serie de criterios para unificar los formatos:

- Los nombres de las tablas se definirán en plural
- La clave primaria de cada tabla será el nombre de la tabla en singular
- Toda la nomenclatura se ha trasvasado a una nomenclatura en letra minúscula y sin tildes para que el software no encuentre ningún problema
- Las columnas que son clave primaria y foránea llevarán como prefijo “cod”, excepto en la tabla “visitas” donde la clave foránea de la tabla “recintos” se definirá como “idrecinto”
- Para definir la procedencia de los datos se ha optado por añadir a las columnas “pre” en referencia a antes de la carga de los datos y “pac” para definir que esa información la ha proporcionado la declaración de la PAC

El resultado de aplicar estos criterios nos ha dejado columnas con diferentes nombres a los anteriores, por un lado, para poder referenciar las columnas con el mismo nombre y por otro lado, para reflejar en el nombre de la columna la información que contiene de forma más representativa. Es el ejemplo de las dos tablas principales, a las que se les ha cambiado el nombre pasando a ser “recintos” (antes SigPac) y “visitas” (antes SITUA).

Además de cambiar diferentes aspectos en el modelo antiguo, también se han creado nuevas tablas para proporcionar a los técnicos un sistema donde puedan controlar los diferentes movimientos que suceden en la base de datos: Estas son las nuevas tablas:

- “recintos_hist”: esta tabla se ha creado para poder almacenar los recintos que de un año a otro ya no existen en la tabla recintos; se puede denominar como un almacén “histórico” de los mismos. Cada año se renuevan tanto el *shapefile* como la información del SigPac, por lo tanto, habrá recintos que se hayan modificado, dejando anticuados algunos que existían en la tabla “recintos”. Estos serán los que se envíen a esta tabla, para que los técnicos puedan consultar esta información cuando les resulte necesario, en vez de eliminarlos.
- “visitas_hist”: este caso es parecido al anterior, ya que esta tabla almacenará registros que se eliminarán de año en año en la tabla “visitas”. Por lo tanto, toda esa información que los técnicos añaden a cada recinto y que de un año a otro se borra totalmente de la tabla “visitas”, tendrá su almacén en esta tabla.
- “control”: por último, esta tabla tiene una función diferente a las anteriores. Su creación se debe a controlar en qué fecha se ha cargado la información a la base de datos. El código de identificación de cada año y cada periodo se

insertará en las tablas “recintos”, “visitas” y en las “históricas”, para poder proporcionar a cada registro la fecha en que se cargó al esquema.

Otra diferencia que existe entre los dos modelos son las diferentes columnas que componen cada tabla. En el modelo antiguo, es decir, en el que utilizaban los técnicos, la organización de todas ellas ha sido cambiada. Este cambio ha sido a veces propuesto por los mismos técnicos, y otras veces, se les ha propuesto a los técnicos cambiarlo para una mejor construcción del modelo definitivo. Estos son los cambios realizados (no todas las tablas han sufrido cambios):

La tabla más extensa como es “visitas” ha perdido algunas columnas haciendo más reducido su volumen de datos (hecho que es conveniente para trabajar de manera más ágil en el sistema). Algunas de esas columnas se han dejado únicamente en otras tablas, ya que en el modelo antiguo se podía encontrar información duplicada (misma columna en diferentes tablas (la razón por la que se trabajaba así era que se debía trasvasar toda la información al archivo .DBF)).

9. Tabla: Diferencia entre “SITUA” y “visitas”

| “SITUA” | | “visitas” |
|---------------|--|---------------|
| JOIN | | idvisita |
| Provincia | | idrecinto |
| Municipio | | regado |
| Poligono | | soa |
| Parcela | | trufa |
| Recinto | | iv |
| SupCat | | supiv |
| SDec | | codforzado |
| CG | | precarga |
| CodigoP | | productopre |
| CodigoV | | suprevista |
| Epoca | | fechaprecarga |
| AñoP | | garantepre |
| Dudoso | | codpac |
| Improductivo | | productopac |
| Abandonado | | variedadpac |
| FechaV | | supac |
| Garante | | codgrupo |
| Observaciones | | codproducto |
| Fase | | codvariedad |
| USO | | codepoca |
| Riego | | superficie |
| Regado | | codgarante |
| CP | | codfase |
| SOA | | fechav |
| Trufa | | codabandono |
| IV | | dudoso |
| SupIV | | improductivo |
| Forzado | | fechap |
| Precarga | | observaciones |
| CodPrevisto | | codcontrol |
| Sprev | | |
| FechaPrecarga | | |
| Garantepre | | |
| PAC | | |
| CodPac | | |
| VarPac | | |
| SPac | | |

La siguiente diferencia entre tablas, corresponde a la tabla “recintos”. En ella, como ya se ha comentado anteriormente se ha integrado la columna para representar los polígonos (“geom”) y se ha creado un índice para la tabla, sustituyéndolo por el “JOIN” anterior. Como se puede observar la mayoría de columnas se mantienen, aunque se han borrado algunas por petición de los técnicos del Departamento. Por último, cabe destacar la inserción de la columna “codcontrol”.

10. Tabla: Diferencia entre “SigPac” y “recintos”

| “SigPac” | | “recintos” |
|------------|---|------------|
| JOIN | | idrecinto |
| Provincia | | geom |
| Municipio | | municipio |
| Poligono | | poligono |
| Parcela | | parcela |
| Recinto | → | recinto |
| Riego | | riego |
| Pend | | coduso |
| CP | | supcat |
| USO | | cp |
| Cap_prev | | codcontrol |
| Cap_manual | | |
| Cap_auto | | |
| OB | | |
| Incid | | |
| Region | | |
| SupSP | | |

Aparte de las tablas principales, las tablas denominadas de “catálogo” también cambiarán en el nuevo modelo. En él, la tabla “grupos” se reduce y ahora solo contendrá un código de cada grupo y su descripción. Las columnas que la acompañan antes pasarán a la tabla “fases”.

11. Tabla: Diferencia entre “Grupos” y “grupos”

| “Grupos” | | “grupos” |
|----------|---|----------|
| CG | | codgrupo |
| Grupo | → | grupo |
| Fase0 | | |
| Fase1 | | |
| Color | | |

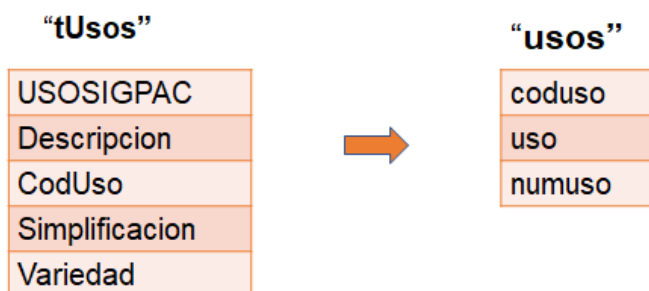
Las columnas que se han eliminado (“revisión” y “color”) en “grupos” se han añadido en “fases” como se ha mencionado, ampliando la tabla del anterior modelo.

12. Tabla: Diferencia entre “Fases” y “fases”



Por último, para terminar con los cambios más notables en el nuevo modelo, nos encontramos con la tabla “usos”. En este caso, se ha reducido el número de columnas por petición de los técnicos, y se han dejado únicamente las que pertenecen a esta tabla (las columnas de “Simplificación” y “Variedad” se definían como claves foráneas).

13. Tabla: Diferencia entre “tUsos” y “usos”



Por otro lado, la importación a la base de datos ha leído ciertos campos con otro tipo de datos, es decir, ha sido un cambio usual cambiar columnas de tipo entero por el tipo “booleano”, ya que la importación la ha leído como entero en vez de como “booleano”. Sin hacer dichos cambios, la relación entre diferentes tablas nos ha problemas ya que no se pueden unir campos de diferentes tipos de datos.

Por último, cabe destacar que se ha añadido una columna denominada “fecha” para poder controlar el día en que se ha trasvasado cada registro de las principales a las tablas “históricas”.

En conclusión, el modelo intermedio ha mostrado las diferencias entre los dos modelos. Éstas guardan relación con el cambio de nomenclaturas, cambio en la organización interna de algunas tablas y adición de algunas tablas para controlar los procesos de trasvases de registros entre las tablas de la base de datos. Tras haber realizado todos estos cambios, el modelo final está preparado para comenzar a trabajar con todos los aspectos de la base de datos unificados para un funcionamiento óptimo

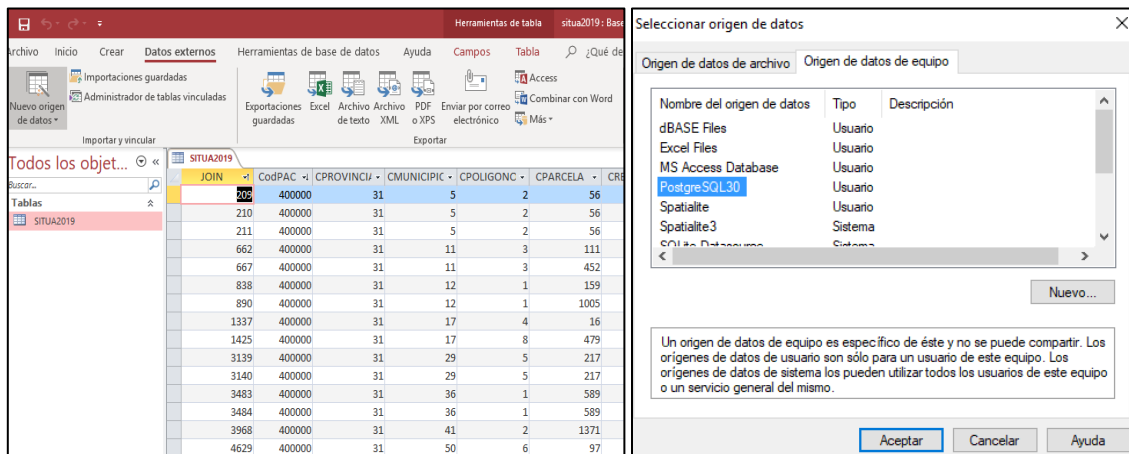
5.3 Proceso de carga, verificación e inserción

Proceso de carga inicial

El proceso que se ha llevado a cabo para cargar la información de los formatos utilizados por los técnicos (archivos en formato Access) a la base de datos de PostgreSQL ha tenido diferentes alternativas. La posibilidad que PostgreSQL brinda para importar información fácilmente contrasta con algunos errores que se crean a la hora de realizar la importación desde Access.

Esta importación, que permite realizarla en tres formatos diferentes como son csv, texto o binario, obliga a transformar la información que se encuentra en tablas Access a cualquiera de estos formatos mencionados. Algunos de los errores muestran son mala carga a PostgreSQL, ya que palabras con letras mayúsculas, minúsculas, con tilde y los diferentes tipos de columnas que lee la nueva base de datos no es la misma que la de origen, por lo que ha generado buscar otra alternativa para el proceso de carga a PostgreSQL.

La alternativa ha sido instalar el “Driver ODBC” (Open DataBase Connectivity). Con ello, mediante la opción que facilita Access para exportarlo a base de datos ODBC se ha podido realizar el proceso sin ningún problema. Tras seleccionar esta opción en Access se debe escoger la base de datos a la que se quiere exportar; en nuestro caso, se ha elegido PostgreSQL. Este trasvase de información se ha realizado para cargar las 14 tablas que se encontraban en la base de datos Access. En esta siguiente imagen se puede observar el proceso:



29. Figura: Proceso de carga a PostgreSQL

Por otro lado, también se ha importado el archivo *shapefile* a la nueva base de datos. Sin embargo, no se ha seguido el mismo proceso que las tablas, ya que el formato en el que se encuentra es diferente.

Para ello, se ha utilizado QGIS, que conectando a la base de datos nos ha permitido, mediante su Administrador, importarlo. En el administrador de la base de datos, se debe definir el archivo que se quiere importar (el *shapefile*) y especificar una serie de características como el SRID de destino, crear geometrías monoparte en vez de multiparte etc.

Tras finalizar el proceso, en la nueva base de datos se habrá creado una nueva tabla con el nombre especificado. La característica especial de esta tabla consiste en que contiene una columna de tipo geometría, que será la que nos permitirá dibujarla en QGIS más adelante.

Una vez cargada toda la información necesaria de un tipo de base de datos a otra este proceso queda finalizado. En él se ha cargado en un esquema temporal, para luego tras efectuar diferentes cambios en un modelo intermedio, trasvasarlo a un nuevo esquema que será el definitivo.

Toda la información volcada a PostgreSQL se ha trabajado mediante el mismo PostgreSQL y también se ha conectado esta misma base de datos desde DBeaver, otra herramienta de administración de base de datos, que es muy intuitiva y fácil de manejar. Cabe destacar que hay diferentes procesos, que PostgreSQL permite realizar de forma más sencilla, pero ha habido otros donde DBeaver lo ha permitido tras comprobar que PostgreSQL ponía alguna restricción.

Verificación

Esta carga “en bruto” de la información adquirida, se ha realizado sin comprobar en detalle sus características. Este proceso de verificación se ha de realizar para poder trasvasar al esquema definitivo, información de calidad, sin redundancias y sin ningún otro tipo de error. Este proceso se ha aplicado a las tablas que no son de “catálogo”. Las tablas de catálogo se han trasvasado al esquema “proyecto”, únicamente aplicándoles los cambios especificados en el modelo intermedio.

La primera verificación que se ha realizado ha sido comprobar que existen los mismos registros en la tabla “SIGPAC” y en la tabla del *shapefile* denominado “muestra”. Es decir, se examinado mediante una consulta si todos los recintos que se encuentran en la tabla se dibujan en el visor y al revés, si todos los recintos visualizados en el visor se encuentran registrados en la tabla mencionada.

Para ello, se ha construido la siguiente consulta. En ella se seleccionan todos los registros que cruzando la información entre las dos tablas den como resultado que el identificador de una de ellas sea nulo, es decir, que no exista. La referencia catastral con los campos municipio, polígono, parcela y recinto son por los que se cruza la información entre registros:

```
select *
from carga.sigpac left join carga.muestra
on carga.sigpac."CMUNICIPIO" = carga.muestra."cmunicipio" and
carga.sigpac."CPOLIGONO"= carga.muestra."cpoligono" and
carga.sigpac."CPARCELA"= carga.muestra."cparcela" and
carga.sigpac."CRECINTO"= carga.muestra."crecinto"
where carga.muestra."id" is null;
```

El resultado de esta consulta no debe mostrar ningún valor; esto significará que todos los valores de la tabla “SIGPAC” se encuentran en la tabla de la capa “muestra”.

Esta comprobación también se debe realizar al revés, para saber si lo que se visualiza se encuentra registrado en la tabla:

```
select *
from carga.muestra left join carga.sigpac
on carga.sigpac."CMUNICIPIO" = carga.muestra."cmunicipio" and
carga.sigpac."CPOLIGONO"= carga.muestra."cpoligono" and
carga.sigpac."CPARCELA"= carga.muestra."cparcela" and
carga.sigpac."CRECINTO"= carga.muestra."crecinto"
where carga.sigpac."JOIN" is null;
```

Si las comprobaciones realizadas dan el mismo número de registros en las dos tablas y cada registro de una tabla se puede encontrar en la otra, nos hemos asegurado que la información guarda un mínimo de calidad y no tendremos ningún error a la hora de integrar los datos de la columna de geometría con las de los registros de la tabla "SIGPAC".

Por otro lado, se ha observado que al importar el *shapefile*, la columna de tipo geometría ha leído los polígonos como "Multipolígonos". Este suceso nos ha hecho dudar de la información del *shapefile*, ya que el sistema de información de la PAC está basado en recintos (la parte más reducida), es decir, cada polígono solo contiene un único polígono.

Esta sospecha nos ha llevado a realizar diferentes consultas para poder examinar si el hecho de mostrar la columna como Multipolígono es un hecho por defecto del programa, o si el *shapefile* contiene algún polígono con más polígonos en su interior.

En la consulta realizada, se ha utilizado la función "ST_NumGeometries" para poder comprobar el número de geometrías de cada recinto; en nuestro caso, se le ha condicionado, poniendo si hay algún recinto con más de una geometría integrada:

```
select * from carga.muestra
where ST_NumGeometries(geom) > 1
limit 100;
```

El resultado ha mostrado un registro:

| 123 id | geom | 123 cprovincia | 123 cmunicipio | 123 cpoligono | 123 cparcela | 123 crecinto |
|---------|---|----------------|----------------|---------------|--------------|--------------|
| 156,852 | MULTIPOLYGON (((556658,4464556117 4707834,9713802 | 31 | 47 | 3 | 324 | 1 |

Este registro corresponde a una parcela en el municipio de Bargota (municipio 47), donde aparece una parcela con diferentes polígonos dentro de ella. Este error, hace que la información sea redundante en la base de datos, ya que los polígonos que se encuentran dentro de la parcela se cuentan dos veces: una de ellas como superficie dentro de la parcela general y la segunda vez, individualmente fuera de la parcela. Al ser un hecho aislado, se ha supuesto que es un error del *shapefile* proporcionado.

Se ha examinado cuantos polígonos construyen esa parcela mencionada mediante la función ST_Dump, que divide la parcela seleccionada en todos los polígonos que contiene:

```
select st_dump(geom) from carga.muestra  
where cmunicipio = 47 and cpoligono = 3 and cparcela = 324;
```

El resultado ha mostrado cinco polígonos:

| path | geom |
|------|--|
| {1} | 0103000020E6640000010000007201000077D495E4E4FC2041E6172BBE7EF551417D9C19B9E1FC2041AAB145AE80F5514170ED81D1D5FC |
| {2} | 0103000020E6640000010000000E0000007A3F86CECBFC20414E77EFD5C8F55141D5D8AFC1D8FC20412AA6BA61C9F55141295FFADEE1F |
| {3} | 0103000020E6640000010000002E00000010382100C7FC204134CB8696D2F55141C8212B80D8FC204143491EADD3F55141E9F42599E0FC2 |
| {4} | 0103000020E664000001000000180000006AB15349F1FB2041B90FB03BE8F55141A9CB210E02FC20415BB3997EE8F55141E8E7672909FC20 |
| {5} | 0103000020E6640000010000003A000000414480043BFB20415E877D230DF65141574CC2CF4FFB20415CCCC6530DF65141433D46CF61FB |

Como ya se ha mencionado en las líneas anteriores, los polígonos 2,3,4 y 5 son redundantes, por lo que de esta parcela solo nos interesa utilizar el primero de ellos para poder trasvasar al esquema definitivo (denominado “proyecto”) una tabla donde la columna geometría sea de tipo “Polígono”.

La solución pasa por escoger los primeros polígonos de todos los recintos de Navarra. Para ello, se ha utilizado la función ST_Geometryn, donde se le pasan unos parámetros con la columna de tipo geometría y el número que se quiere escoger. Esta función se ha aplicado a la siguiente consulta, donde se crea una nueva tabla en el esquema “carga”, añadiendo a la tabla “SigPac” la columna de geometría para poder representar los polígonos.

```
create table carga.sigpacgeom as  
select "CPROVINCIA", "CMUNICIPIO", "CPOLIGONO", "CPARCELA", "CRECINTO",  
"RIEGO", "USO", "SUPSP", "JOIN", "PRODUCTO", "FASE", st_geometryn(geom, 1) geom  
from carga.sigpac A inner join carga.muestra B  
on A."CMUNICIPIO" = B."cmunicipio" and  
A."CPOLIGONO" = B."cpoligono" and  
A."CPARCELA" = B."cparcela" and  
A."CRECINTO" = B."crecinto"
```

Inserción

Tras verificar la información cargada al esquema inicial, la situación es la siguiente: en el esquema “carga” se encuentran las tablas “sigpacgeom” y “situa”, y en el esquema “proyecto” se encuentran las tablas de “catálogo”.

El siguiente paso consistirá en crear vacías las dos tablas “principales” (“recintos” y “visitas”) mediante consultas, escogiendo los campos que deben tener (indicado por los técnicos del Departamento), el tipo de cada campo, las correspondientes claves y algunos índices para facilitar diversos procesos. Estas consultas se pueden ver en el apartado Anexos.

Una vez creadas las tablas, debemos insertar los datos que proceden de las tablas “sigpacgeom” y “situa”. Para rellenar la tabla “recintos” se ha realizado la siguiente consulta; en ella se han escogido los campos necesarios de la tabla que contiene los datos (“sigpacgeom”) y se han insertado en la tabla de destino (“recintos”). En la columna “codcontrol” se ha asignado el código correspondiente a la fecha y periodo correspondiente de carga:

```
insert into proyecto.recintos(cmunicipio, cpoligono, cparcela, crecinto,
riego, coduso, supcat, geom,codcontrol)
select A."CMUNICIPIO", A."CPOLIGONO", A."CPARCELA", A."CRECINTO",
A."RIEGO", A."USO", A."SUPSP", A."geom",1
from carga.sigpacgeom A
```

El mismo proceso se ha llevado a cabo con la tabla “visitas”. En este caso la información procederá de la tabla “situa”. Ésta es bastante más amplia que la anterior tabla, por lo que la consulta también se alarga al escoger todos los campos requeridos para el nuevo registro. Como en el caso anterior, se ha insertado también información acerca del control de carga a la nueva tabla.

```
insert into proyecto.visitas(idrecinto,regado,soa,trufa,iv,supiv,codforzado,
precarga,productopre,suprevista,fechaprecarga,garantepre,codpac,productopac,
variedadpac,supac,codgrupo,codproducto,codvariedad,codepoca,superficie,codgarante,
codfase,fechav,codabandono,dudoso,improductivo,fechap,observaciones,codcontrol)
select a.idrecinto, B."REGADO",B."SOA",B."Trufa",B."IV",B."SupIV",B."Forzado",
B."Precarga",B."CodPrevisto",B."SPrev",B."FechaPrecarga",B."GarantePrecarga",B."PAC",
B."CodPAC",B."VarPAC",B."SPAC",B."CG",B."CódigoP",B."CódigoV",B."Epoca",B."SDec",
B."Garante",B."Fase",B."FechaV",B."Abandono",B."Dudoso",B."Improductivo",B."AñoP",B."OBS",1
from proyecto.recintos A inner join carga.situa B
on A."cmunicipio"=B."CMUNICIPIO" and A."cpoligono"=B."CPOLIGONO" and
A."cparcela"=B."CPARCELA" and A."crecinto"=B."CRECINTO";
```

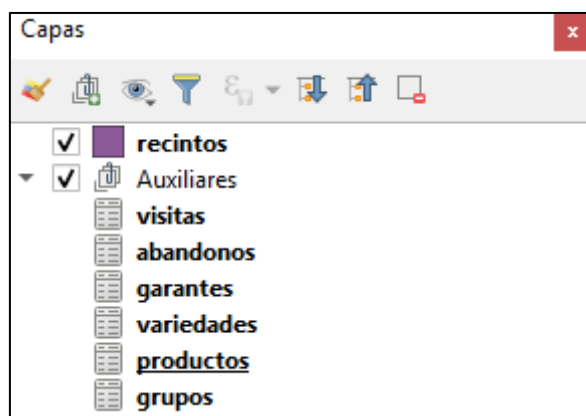
El proceso de carga inicial se da por terminado en este punto. El estado de la base de datos es: el esquema “proyecto” contiene la información necesaria para poder empezar a trabajar, con las tablas “recintos y “visitas” rellenas y las tablas “catálogo”, proporcionando los códigos necesarios.

5.4 Sistema

En el apartado anterior se han dejado las tablas acondicionadas para poder empezar a trabajar desde QGIS. Para ello, se deberá conectar mediante el “Administrador de Base de Datos” la base de datos diseñada con el proyecto de QGIS en el que se trabajará. Esta conexión se ha explicado a detalle en el apartado de “Manual de usuario”.

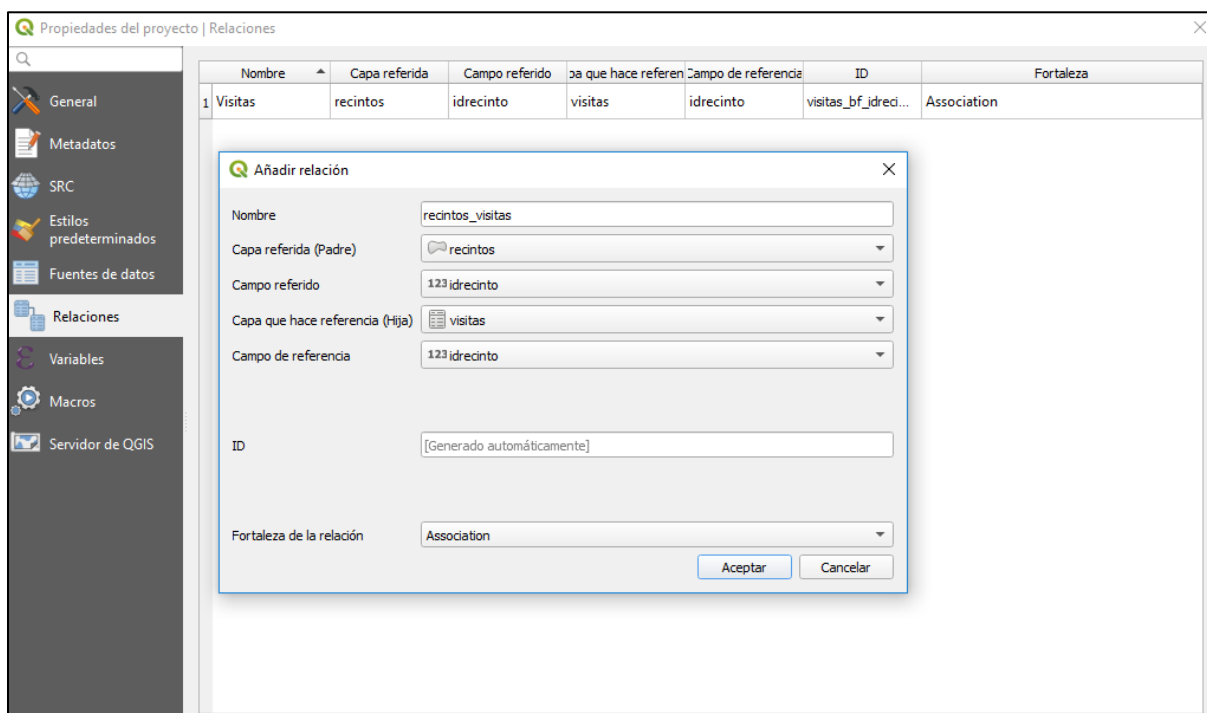
El primer paso consistirá en cargar las capas al proyecto; en nuestro caso tenemos una única que se puede representar: “recintos”. Además de ella, se han cargado también las tablas auxiliares. Cabe destacar que solo se han escogido las tablas con

aquellos campos que interesan que aparezcan en el formulario. Este formulario mencionado, será el que los técnicos utilizarán para introducir información.



30. Figura: Imagen del menú de capas cargadas

Tras cargar todo lo necesario, se ha realizado una relación entre “recintos” y “visitas” como se ha hecho en el modelo. Esta relación es 1:N (uno a muchos) y nos permitirá enlazar el recinto que visualizamos con los apuntes que queramos introducir sobre él. En “propiedades del proyecto” se ubica la herramienta para realizar lo mencionado, donde el campo de enlace será “idrecinto”, ya que en la tabla “visitas” este campo juega el papel de clave foránea para poder construir dicho enlace.

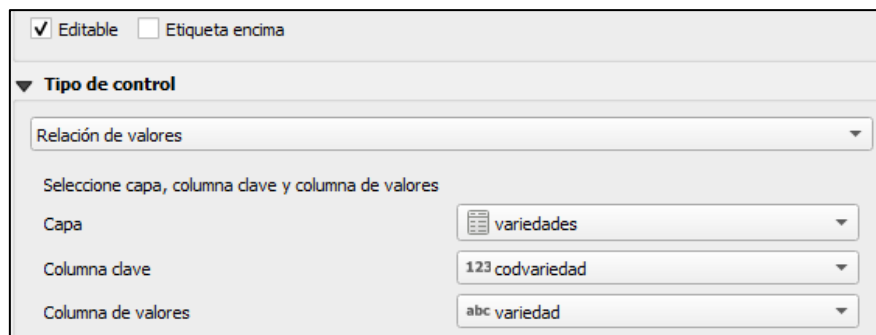


31. Figura: Construcción de la relación recintos-visitas

El siguiente paso corresponderá a la confección del formulario. Éste se ha construido con los mismos campos y con la misma configuración que aparecían en el formulario del visor antiguo.

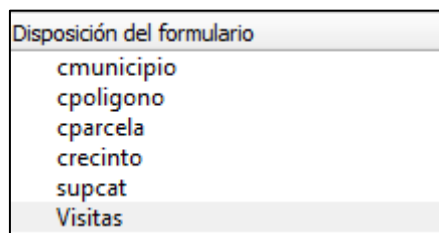
En las “propiedades de capa” de la tabla “visitas”, en la pestaña “formulario” podemos confeccionar aquellos campos que son de esa misma tabla y, además, los campos que son clave foránea enlazando las diferentes tablas auxiliares o de catálogo.

Así, lo que se ha configurado en este menú ha sido la opción de que sea editable y el tipo de control para cada campo. En general, se han asignado los controles de con esta configuración: a columnas de tipo booleanos se ha asignado “casilla de verificación”, a columnas numéricas “edición de texto” y en columnas como “productos”, “variedades” o “abandonos” se ha escogido el control “relación de valores” para poder escoger una opción de la lista. En esta imagen se observa el proceso de configuración del campo “variedad”. Aunque el campo de enlace sea su código (codvariedad) interesa que aparezca la descripción de ese código (columna denominada “variedad” dentro de la tabla “variedades”). Este proceso señalado se repetirá para todos los campos requeridos.



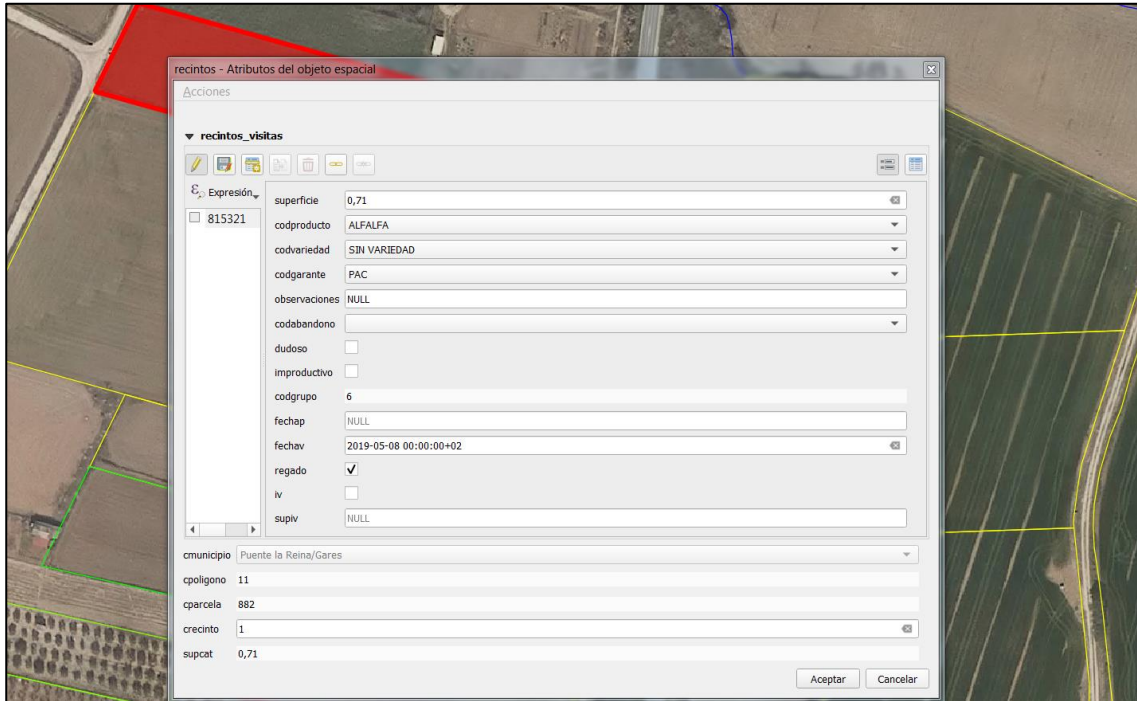
32. Figura: Configuración de los campos del formulario

Esta configuración realizada en la pestaña “formulario” de la tabla “visitas” debe de tener su enlace en la misma pestaña, pero de la tabla “recintos”. Para ello, se introducen en el formulario los campos requeridos de la tabla “recintos” (municipio, polígono, parcela, recintos y la superficie catastral) y se debe añadir la configuración realizada en la tabla “visitas” (denominada en este caso: “Visitas”).



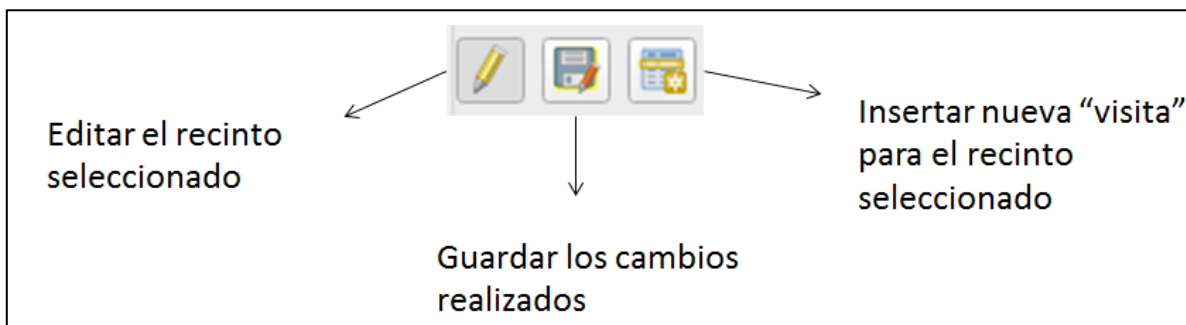
33. Figura: Disposición final del formulario

El resultado de todo este proceso se visualizará al identificar un recinto cualquiera en la interfaz de QGIS: Al realizar dicha selección, automáticamente se abrirá una ventana visualizando el formulario con la configuración determinada anteriormente.



33.Figura: Visualización del formulario de un recinto en pantalla

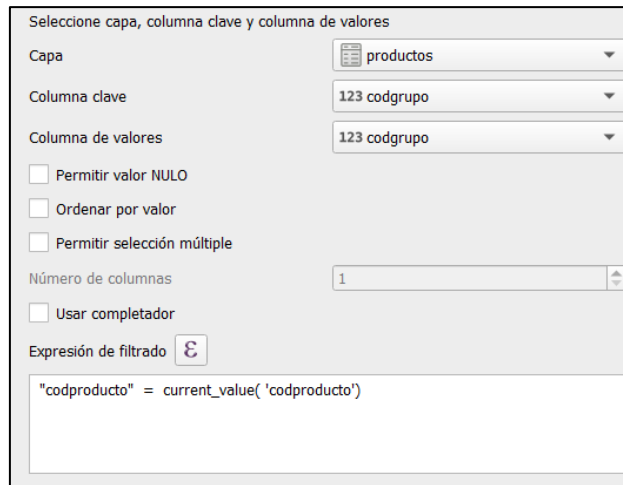
En esta ventana se puede visualizar la información en la parte de arriba de la tabla “recintos” y debajo de ella la información de los campos enlazados con la tabla “visitas”. En este editor de 1:N se puede agregar todas las visitas que se quieran con el siguiente botón o editar la información existente:



35. Figura: Opciones del formulario para la edición de atributos

En este formulario, los técnicos necesitan varios procesos automáticos en los que tras actualizar un campo se debe de cambiar otro al instante y automáticamente. Para ello, en la configuración de los campos se han introducido “expresiones de filtrado” en las que los valores de relación toman un papel muy importante. Esta

siguiente imagen muestra un ejemplo de las “automatizaciones” que se han diseñado cuando los técnicos actualicen los campos del formulario:



Selección de capa, columna clave y columna de valores

Capa: productos

Columna clave: 123 codgrupo

Columna de valores: 123 codgrupo

Permitir valor NULO

Ordenar por valor

Permitir selección múltiple

Número de columnas: 1

Usar completador

Expresión de filtrado: `"codproducto" = current_value('codproducto')`

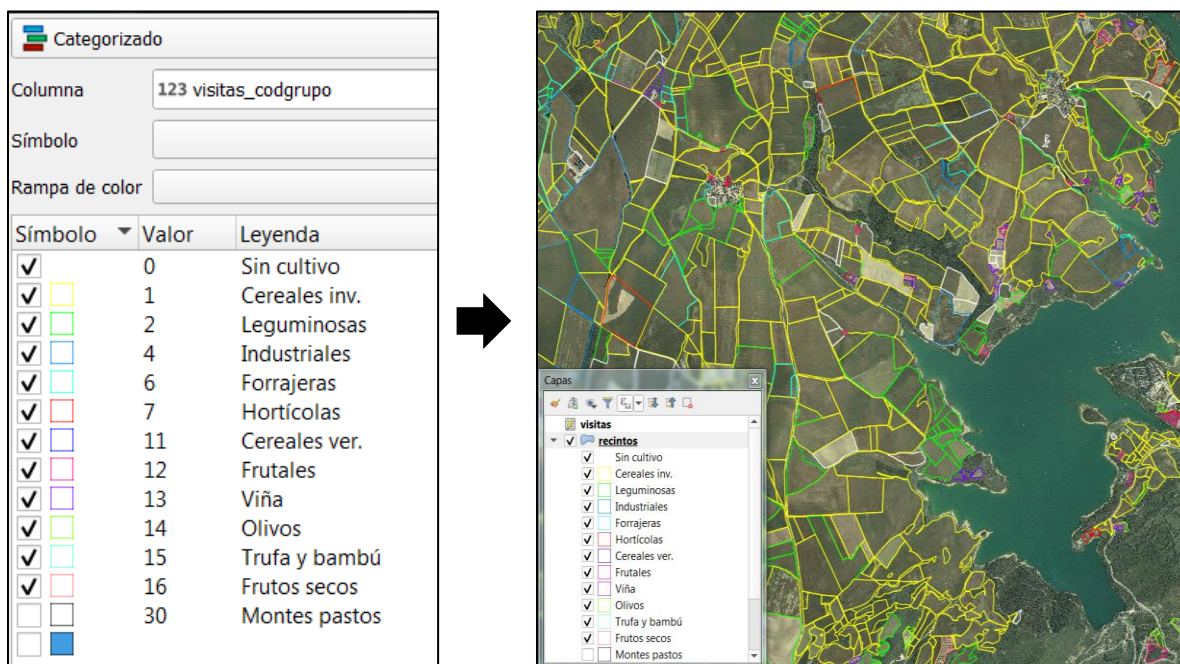
36. Figura: Expresiones de filtrado para automatizar procesos en el formulario

Tras haber enlazado la información requerida y diseñado el formulario, el siguiente paso es adecuar la simbología de la capa “recintos”. Esta adecuación se ha realizado para tratar de asemejar a la simbología con la que trabajan los técnicos en el anterior visor.

Hay que tener en cuenta que la tabla “fases” es la que posee el campo denominado “color”. Este campo nos servirá para identificar cada grupo de cultivos, por ello podemos encontrar en la misma tabla la clave foránea que une las tablas “fases” y “grupos” (“codgrupo”).

Para poder reflejar el color de cada cultivo en el visor debemos de unir las tablas “recintos” y “visitas”, ya que esta última contiene un campo con código de grupo (“codgrupo”) por la que podemos acceder al código de cada grupo de cultivos. A cada grupo, se le asignará el color que se especifica en la tabla “fases”.

El objetivo de este proceso era poder implementar una simbología categorizada para la capa “recintos” mediante el campo “codgrupo” que se sitúa en la tabla “grupos”. La simbología escogida se caracteriza por un relleno transparente con el borde de cada recinto coloreado. Con ello, se puede visualizar los diferentes colores de cada recinto que proporciona la imagen añadida al visor y al mismo tiempo ver el color del grupo de cultivo de la declaración de ese periodo.



37. Figura: Elección de la simbología por cada grupo de cultivos

Por lo tanto, mediante este formulario podremos editar la información de la base de datos estando en la oficina o si los técnicos salen al campo (en este caso, se deberá conectar a un GPS. Este proceso está reflejado en el “Manual de usuario”). Además, la simbología utilizada está asignada de la misma manera para cada grupo de cultivos.

5.5 Actualización

Tras haber realizado la carga inicial a la base de datos y una vez, trabajado con los datos cargados, los técnicos del Departamento necesitan actualizar la base de datos, ya que anualmente se debe renovar la información, por los posibles cambios que se puedan dar. No solo anualmente, la declaración de la PAC, se testifica dos veces al año, por lo cual, la base de datos debe de ser actualizada durante un mismo curso también.

Así, se ha construido un sistema de consultas para la actualización de la información. Cabe destacar que, en este proceso de actualización, habrá información anticuada que será eliminada de las tablas que se emplearán para trabajar. En este sistema diseñado, esta información no se eliminará definitivamente, si no que se almacenará en las tablas denominadas “históricas” (“recintos_hist” y “visitas_hist”) para que los usuarios del sistema puedan consultar dicha información cuando sea necesario.

Los primeros pasos de este proceso serán idénticos al del proceso inicial de carga. Los técnicos recibirán información acerca de los recintos de “SigPac” y también se harán con nuevo *shapefile*. Estos dos archivos se deben subir al esquema “carga” mediante la conexión Access-PostgreSQL y QGIS, respectivamente. Una vez en la

base de datos, se realizarán las respectivas verificaciones para después unir las dos tablas en la denominada tabla “sigpacgeom”.

Para actualizar las tablas “principales” de la base de datos, se han creado diversas consultas que vamos a proceder a explicar.

La tabla “visitas” recogerá a lo largo del año todas las observaciones que los técnicos hayan recogido apoyándose en las diversas fuentes de información. Esta información no será válida al siguiente año, por lo que deberá ser vaciada al término del año. Para ello, primero debemos mandar toda la información a la tabla “visitas_hist”, con lo que quedará registrado en qué fecha pasó a esta última tabla (esto se consigue empleando la función “NOW()”). Tras realizar ese trasvase, la tabla deberá ser vaciada.

```
insert into proyecto.visitas_hist(idvisita,idrecinto,regado,soa,trufa,iv,supiv,codforzado,
precarga,productopre,suprevista,fechaprecarga,garantepre,codpac,productopac,
variedadpac,supac,codgrupo,codproducto,codvariedad,codepoca,superficie,codgarante,
codfase,fechav,codabandono,dudoso,improductivo,fechap,observaciones,codcontrol,fecha)
select idvisita,idrecinto,regado,soa,trufa,iv,supiv,codforzado,
precarga,productopre,suprevista,fechaprecarga,garantepre,codpac,productopac,
variedadpac,supac,codgrupo,codproducto,codvariedad,codepoca,superficie,codgarante,
codfase,fechav,codabandono,dudoso,improductivo,fechap,observaciones,codcontrol, NOW()
from proyecto.visitas
truncate table proyecto.visitas;
```

La tabla “visitas” por lo tanto quedará vacía tras ese proceso. Hay que destacar que el orden que se acata en este proceso no es aleatorio, si no que se lleva a cabo de forma que el sistema no de ningún problema (este paso explicado previamente al realizarse después del siguiente que se va a explicar daría error, ya que las claves definidas no dejarían borrar el contenido de la tabla).

La actualización de la tabla “recintos” se puede considerar más compleja, ya que este proceso contiene más pasos (cuatro exactamente) en los que se debe comparar la tabla que contiene la información nueva (“sigpacgeom”) con la ya existente (“recintos”).

1. Trasvasar la información antigua: al hacer la comparación, habrá registros en la tabla “recintos” que ya no existen en la información nueva que contiene “sigpacgeom” (hay ciertos recintos que desaparece su referencia catastral porque se hayan realizado diferentes concentraciones parcelarias o se ha cambiado la distribución de la parcela etc). Por lo tanto, estos registros “anticuados” pasarán a la tabla histórico de recintos, quedando registrada también la fecha en que se ha realizado este proceso.

```
insert into proyecto.recintos_hist(idrecinto,geom,cmunicipio,cpoligono,
cparcela,crecinto,supcat,coduso,riego,cp,codcontrol,fecha)
select R.idrecinto,R.geom,R.cmunicipio,R.cpoligono,
R.cparcela,R.crecinto,R.supcat,R.coduso,R.riego,R.cp, R.codcontrol, NOW()
from proyecto.recintos R left join carga.sigpacgeom S
on S."CMUNICIPIO" = R."cmunicipio" and
S."CPOLIGONO" = R."cpoligono" and S."CPARCELA" = R."cparcela" and S."CRECINTO" = R."crecinto"
where S."JOIN" is null;
```

2. Eliminar la información trasvasada: aunque se haya pasado la información a la tabla “recintos_hist”, estos registros trasvasados continúan en la tabla “recintos” por lo que debemos eliminarlos. En la consulta se ha hecho un filtro para eliminar solo los que se han trasvasado en el periodo que queremos (columna “codcontrol”).

```
delete from proyecto.recintos
where recintos.idrecinto in
(select idrecinto
from proyecto.recintos_hist
where recintos_hist.idrecinto=recintos.idrecinto and codcontrol=1 and codcontrol= 2)
```

3. Actualizar los registros existentes: en este proceso de cruce de información entre las dos tablas, puede que haya ciertos registros donde manteniendo la información de “base” (referencia catastral) se cambie la información restante (por ejemplo su geometría (utilizada la función ST_Equals, en este caso en negativo para examinar si el recinto ya no es igual que el anterior), su uso, si posee sistema de riego etc). Para ello se deberá actualizar la información existente.

```
update proyecto.recintos set geom=S.geom, riego=S."RIEGO", supcat=S."SUPSP", coduso=S."USO"
from carga.sigpacgeom S
where S."CMUNICIPIO" = proyecto.recintos."cmunicipio" and
S."CPOLIGONO"= proyecto.recintos."cpoligono" and
S."CPARCELA"= proyecto.recintos."cparcela" and
S."CRECINTO"= proyecto.recintos."crecinto" and
(not ST_EQUALS(S.geom,recintos.geom) or
S."RIEGO"!=recintos.riego or S."SUPSP"!=recintos.supcat or S."USO"!=recintos.coduso)
```

4. Insertar nuevos registros: por último, la información nueva contenida en la tabla “sigpacgeom”, puede tener registros creados ese nuevo año que no se encuentren en la tabla “recintos”, por lo que se deben insertar en el esquema definitivo.

```
insert into proyecto.recintos(idrecinto,geom,cmunicipio,cpoligono,cparcela,crecinto,supcat,riego,coduso)
select S."JOIN",S.geom,S."CMUNICIPIO",S."CPOLIGONO",S."CPARCELA",S."CRECINTO",S."SUPSP",S."RIEGO",S."USO"
from carga.sigpacgeom S left join proyecto.recintos R
on S."CMUNICIPIO" = R."cmunicipio" and
S."CPOLIGONO"= R."cpoligono" and
S."CPARCELA"= R."cparcela" and
S."CRECINTO"= R."crecinto"
where R.idrecinto is null;
```

La fase actualización quedaría completada tras realizar estos cuatro pasos explicados. Por lo tanto, el estado de la base de datos está actualizada con los registros de “recintos” actualizados.

En conclusión, el sistema que se ha implementado contiene una base de datos nueva con nuevo formato (base de datos espacial), donde se carga una información inicialmente cambiando ciertos formatos para una mejor adecuación y se verifica su calidad antes de trabajar con ella.

Tras obtener esa información, los técnicos recogerán cierta información que añadirán a la ya cargada mediante QGIS, donde editarán la base de datos mediante un formulario diseñado a medida. Cada vez que el Departamento obtenga nueva información deberá actualizar la ya existente, mandando la información anticuada a tablas de almacenamiento histórico y actualizando la información de existente. Este sistema se ha diseñado también para dejar preparada la base de datos para el siguiente año y volver a repetir el proceso.

6. Tecnologías empleadas

En este punto se van a describir los diferentes programas y lenguajes de programación utilizados para realizar el presente proyecto.

6.1 Programas

6.1.1 PostgreSQL

PostgreSQL es un sistema gestor de base de datos, posee una sólida reputación por sus características donde se pueden destacar la integridad de datos, la robustez del conjunto de características, extensibilidad y además, al ser un software de código abierto ofrece diferentes soluciones y de alto rendimiento. PostgreSQL se ejecuta en los principales sistemas operativos y cuenta con potentes complementos como el popular extensor de bases de datos geoespaciales PostGIS⁷.

En este caso, este sistema es la base de este proyecto. Se ha utilizado para obtener los datos una vez exportados de Acces; para esta aplicación se utilizará este gestor por sus diferentes ventajas citadas anteriormente.



38. Figura: Icono de PostgreSQL

6.1.2 DBeaver

DBeaver (community edition) es un software que actúa como una herramienta de base de datos universal destinada a desarrolladores y administradores de bases de datos. DBeaver tiene una interfaz de usuario bien diseñada, es una plataforma basada en un marco de código abierto y permite escribir múltiples extensiones, así como también es compatible con cualquier base de datos.

Esta herramienta permite “cambiar” el comportamiento de la aplicación para proporcionar funcionalidades específicas de la base de datos o características que son independientes de la base de datos. Esta es una aplicación de escritorio escrita en Java y basada en la plataforma Eclipse.⁸

En este proyecto, este software se ha utilizado como gestor esencial de la base de datos con el que se ha explotado, se ha construido consultas y se han analizado los datos originales para poder crear el nuevo modelo de datos. Para trabajar con DBeaver se ha tenido que conectar con PostgreSQL.



39. Figura: Icono de DBeaver

6.1.3 QGIS

⁷ <https://www.postgresql.org/about/>

⁸ <https://dbeaver.io/about/>

QGIS (QuantumGIS) es una aplicación profesional de SIG (Sistema de Información Geográfica) que está construida sobre Software Libre y de Código Abierto (FOSS). Es un proyecto oficial de Open Source Geospatial Foundation (OSGeo). Corre sobre Linux, Unix, Mac OSX, Windows y Android y soporta numerosos formatos y funcionalidades de datos vector, datos ráster y bases de datos.

Entre sus características se encuentran dar soporte para la extensión espacial de PostgreSQL: PostGIS, la interconexión con muchas bases de datos geoespaciales (aparte de la ya mencionada) como pueden ser: Geopackage, Spatialite, Oracle etc., y también cuenta con una gran ventaja como puede ser usar QGIS como GUI (*Graphic User Interface*) del SIG GRASS, utilizando toda la potencia de análisis de este último en un entorno de trabajo más amigable.⁹

En este proyecto, QGIS será la herramienta donde se ha montado la nueva aplicación para la recogida de datos, ya que tendrá conexión con la base de datos para poder modificarla según los diferentes cultivos visibles a lo largo de toda Navarra.



40. Figura: Icono de QGIS

6.1.4 PostGIS

PostGIS es la extensión espacial de PostgreSQL, que convierte al sistema de administración de la base de datos en una base datos espacial. Añade soporte para objetos geográficos permitiendo que las consultas de ubicación se ejecuten en SQL.

Las funciones integradas en esta extensión aumentan el potencial de PostgreSQL que en comparación con otras bases de datos espaciales se coloca por encima de todas ellas. Su fácil utilización y la opción de integrar diferentes tipos de datos hacen de PostGIS una extensión muy utilizada a nivel mundial.¹⁰

Esta extensión se ha añadido a PostgreSQL para poder importar el *shapefile* que utilizan los técnicos acerca de los recintos SIGPAC y así, poder integrarlo en otras tablas, integrando datos alfanuméricos y espaciales en una misma tabla.



41. Figura: Icono de PostGIS

6.1.5 Geoserver

⁹ <https://www.qgis.org/es/site/about/index.html>

¹⁰ <https://postgis.net/>

Geoserver es un servidor que publica datos de cualquier fuente de datos espaciales utilizando estándares abiertos, en formatos diferentes y para clientes estándar como navegadores web y programas GIS de escritorio.

Los datos se publican a través de interfaces basadas en estándares, como WMS, WFS, WCS, WPS, Tile Caching y más. En este proyecto, este servidor se ha utilizado para publicar los datos y mapas empleados en el sistema GIS.



42. Figura: Icono de Geoserver

6.2 Lenguajes de programación

Entre los lenguajes de programación que se han utilizado están los siguientes:

6.2.1 SQL

SQL es un lenguaje de acceso a bases de datos relacionales que permite realizar diferentes tipos de operaciones en ellas, tanto como cualquier consulta para su verificación, modificación o edición.

En este trabajo el lenguaje mencionado se ha utilizado para poder insertar información en la base de datos; tanto para actualizarla como para cargarla la primera vez después de su creación. Por otro lado, cualquier asignación a los datos de la misma se ha realizado a través de SQL.



43. Figura: Icono de SQL

7. Manual de usuario

El sistema GIS diseñado en este proyecto requiere de unas instrucciones mínimas para su uso. En este apartado se explicará brevemente la instalación de los softwares necesarios para la utilización del sistema y unas instrucciones para que cualquier usuario pueda trabajar con él.

7.1 Instalación

7.1.1 Instalación de PostgreSQL + Postgis

El primer punto de este apartado corresponde a la instalación del administrador de la base de datos: PostgreSQL. Para instalarlo hay que tener en cuenta el sistema operativo de nuestro ordenador, ya que el instalador no es el mismo para 32bits o para 64.

Para descargar el software el enlace es el siguiente, se recomienda siempre descargarse la última versión disponible en este enlace:

<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads#windows>

Una vez descargado el software en la versión más reciente y seleccionado el sistema operativo se deben seguir ciertas instrucciones:

- Se especificará el directorio de instalación y de los datos
- Habrá que definir un usuario y una contraseña para poder conectar cada vez que abramos la base de datos
- Se indicará un número de puerto en el que el servidor debería escuchar
- En los siguientes pasos de instalación se seleccionará “siguiente” hasta llegar a la pantalla que menciona que se ha finalizado la instalación, donde desmarcaremos la casilla “*Stack Builder*” y pulsaremos “Terminar”.

El segundo paso corresponderá a la instalación de PostGIS, que debemos escoger la misma versión de bits que hayamos utilizado en el paso anterior al instalar el PostgreSQL.

Para descargar el software accederemos a este enlace:

- 32 bits: <http://download.osgeo.org/postgis/windows/pg96/postgis-bundle-pg96x32-setup-2.3.0-1.exe>
- 64 bits: <http://download.osgeo.org/postgis/windows/pg96/postgis-bundle-pg96x64-setup-2.3.2-1.exe>

Para instalar el software se debe ejecutar con permisos de Administrador, antes de seleccionar la opción de “PostGIS”, dejando deseleccionada la opción “*Create spatial database*”.

En el siguiente paso aparecerá el directorio donde se guardará la instalación (por defecto reconoce la que se ha seleccionado anteriormente). Y, finalmente, debemos de seleccionar “sí” los siguientes cuadros de diálogo que nos aparecen.

Para finalizar, se procederá a instalar la herramienta de administración y consulta PgAdmin. Esta herramienta, funciona bien en su tercera versión, pero se recomienda trabajar mediante su cuarta versión:

PgAdmin4: <https://www.postgresql.org/ftp/pgadmin/pgadmin4/v4.11/windows/>

Después de descargar la carpeta comprimida, se debe descomprimirla en una carpeta y ejecutar el instalador (archivo .msi). Las opciones que aparece se dejarán por defecto.

7.1.2 Instalación Geoserver

En este segundo punto se van a definir los pasos a realizar para la instalación de Geoserver. Antes de nada, se debe acceder a este enlace para poder descargar la última versión:

<http://geoserver.org/download/>

Una vez descargado el archivo, se ejecuta y aparecerá una ventana para configurar la instalación. En dicha ventana se determinarán diferentes cuestiones:

- Se deberá aceptar la licencia del producto
- Se define el directorio de instalación
- En la siguiente pantalla se solicita elegir la pantalla de Menú inicio
- El siguiente paso corresponde a seleccionar el directorio que contiene el complemento JRE (Java Runtime Environment)
- Otro directorio que se debe definir es el de la carpeta Data Directory de Geoserver
- En dicho servidor se debe tener cuenta de usuario, por lo que debemos escribir un usuario y contraseña. Además se especificará el puerto al que Geoserver responderá
- El último paso se pide definir el tipo de instalación a realizar: se seleccionará "Instalar como servicio"

Una vez instalado, podremos abrir un navegador y acceder a la interfaz de Administración Web de Geoserver ingresando la siguiente dirección (se debe especificar el localhost de cada usuario):

<http://192.168.1.38:8080/geoserver>

7.1.3 Instalación QGIS y conexión con PostgreSQL

Para la instalación del Sistema de Información Geográfica QGIS, es necesario borrar (si se da el caso) las versiones anteriores del software, ya que suele dar problemas tener instalados dos versiones diferentes del programa en un mismo equipo.

Para acceder a la descarga se debe hacerla mediante este enlace:

<https://qgis.org/es/site/forusers/download.html>

En esa dirección se puede elegir el Sistema Operativo y la versión que se quiere descargar. Después de realizar la descarga el proceso es muy sencillo, ya que solo se debe determinar el directorio donde se quiere instalar y el instalador "autónomo" realiza todo lo demás.

En este trabajo, QGIS está conectado con PostgreSQL para poder editar la base de datos desde la misma visualización de los recintos. Para realizar esta conexión se debe acceder desde QGIS a la pestaña “Base de Datos” que se encuentra en la parte superior del programa. Tras seleccionar “Administrador de Base de Datos” se nos abre una ventana donde la conexión que puede realizar con diferentes Base de Datos, como pueden ser Oracle, Spatialite, PostgreSQL o incluso con el formato novedoso Geopackage.

EN nuestro trabajo se ha trabajado con PostgreSQL, por lo que seleccionaremos esta opción y al instante, nos aparecerá una ventana para indicar el usuario y contraseña. Éstos los hemos definido al instalar el PostgreSQL, por lo que una vez de escribirlos, tenemos conectado el QGIS con la Base de Datos deseada.

Esta conexión permite carga en QGIS las tablas creadas en la base de datos, visualizar las columnas de tipo geometría de la tabla espacial e incluso modificar la Base de Datos desde QGIS, objetivo principal del este proyecto

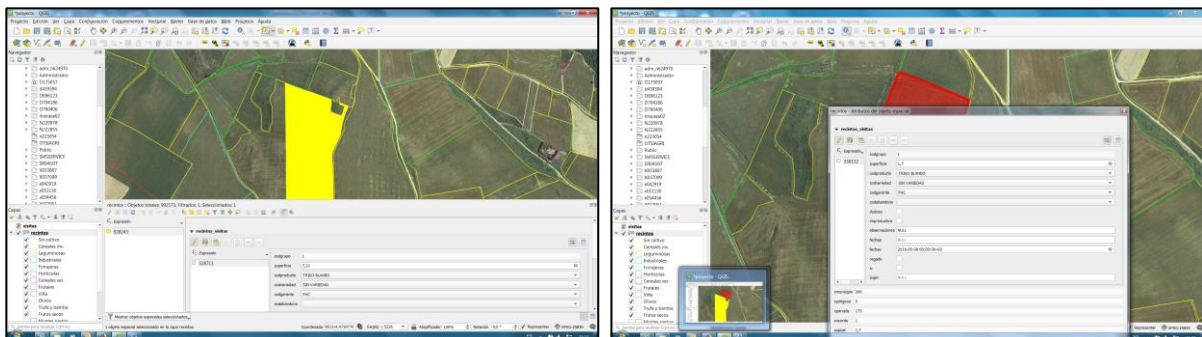
7.2 Uso

7.2.1 Modificación de la base de datos

Tras realizar los anteriores pasos y conectar los diferentes softwares entre sí, se podrá modificar la base de datos desde el mismo QGIS. Para ello, se deben cargar tanto la tabla “recintos” como la tabla “visitas” en QGIS. La primera nos permitirá visualizar cada uno de los recintos de toda Navarra y la segunda, por su parte, será la que posibilitará añadir y modificar los “apuntes” de cada recinto.

Mediante la herramienta de selección, se podrá escoger el recinto deseado. Tras elegirlo, aparecerá una ventana acoplada con el formulario. En dicho formulario aparecerán los campos editables de la base de datos, tras rellenarlos o modificarlos quedarán guardados.

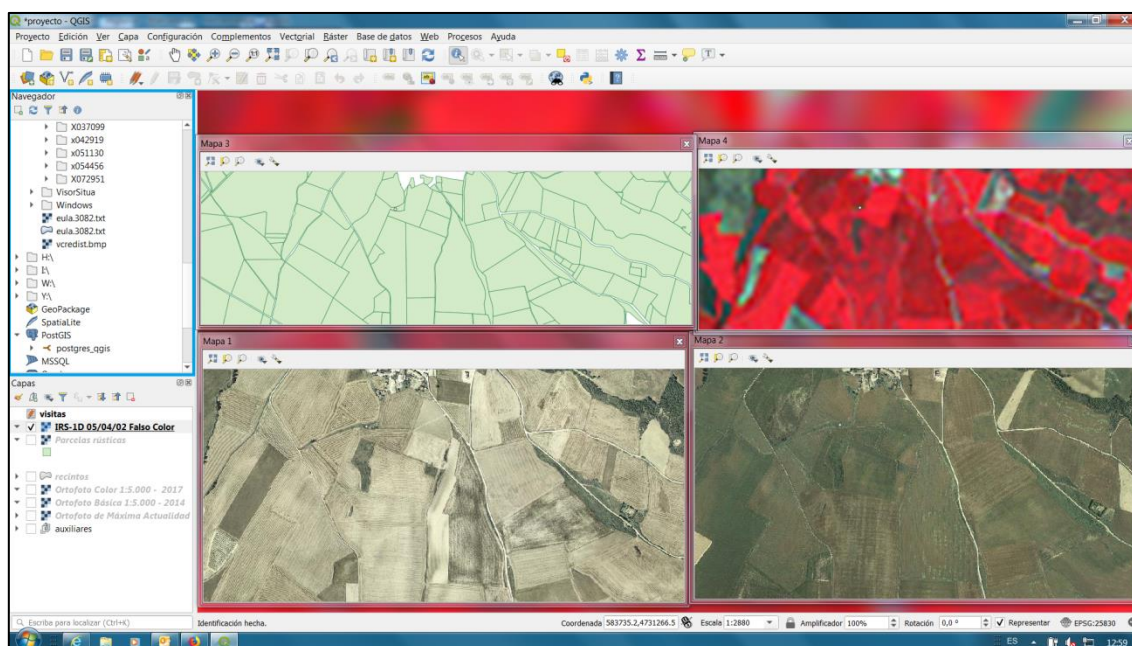
Otra opción para poder editar la base de datos es mediante la opción de “identificación de objetos espaciales”. En esta ocasión, se abrirá una ventana auxiliar con el formulario. Las dos opciones aparecen en esta siguiente figura:



44. Figura: Edición de la Base de Datos mediante la selección del recinto

Para la modificación de datos, los técnicos propusieron la posibilidad de poder comparar diferentes mapas desde el mismo visor. Para ello, QGIS ofrece la

posibilidad de insertar varias vistas de mapa en el canvas, dividiendo la pantalla en tantas vistas como mapas queramos comparar.



45. Figura: División de la pantalla de QGIS

Por otro lado, en la parte izquierda de la pantalla de QGIS, aparecen diferentes herramientas donde el usuario puede escoger cual de ellas visualizar en la interfaz. En este caso, se ha optado por escoger el acceso al “navegador”, ya que los técnicos necesitan tener acceso rápido a las diferentes imágenes base para poder compararlas (recuadro azul en la figura 43).

Esta modificación de los datos se realizará tanto en la oficina como en el campo. En este último caso, se deberá proporcionar señal de un dispositivo GPS mediante bluetooth, ya que las revisiones se realizan en vehículo. A continuación, se definirán los pasos para conectar el sistema a la señal de GPS, para poder asistir al seguimiento del vehículo en vivo.

Para poder visualizar la información de GPS que ofrece QGIS, debemos de acceder a “Ver” y después en “Paneles” seleccionar la opción de “Información de GPS”. Tras ejecutar dicha selección, se abre una ventana con la configuración por defecto de QGIS. En este menú, se pueden seleccionar el tipo de conexión, la digitalización si se quiere dibujar polígonos, diferentes opciones para ver el cursor y el centrado de mapa que será importante a la hora de trabajar desde el vehículo. En nuestro caso, en la pestaña de “conexión” se deberá seleccionar la opción “COM3: Serie estándar sobre el vínculo Bluetooth”. Simplemente al seleccionar el botón de conectar, el cursor aparecerá en la ubicación donde se encuentra el vehículo. Por lo tanto, se podrá comenzar a trabajar.

7.2.2 Publicación de datos

La publicación de los datos, *shapefiles* etc. que se han creado se pueden publicar mediante Geoserver. Para acceder debemos de escribir el usuario y contraseña señalados en los pasos de instalación.

El archivo que se quiere publicar debemos de guardar en el directorio donde se encuentra la carpeta “Geoserver” y dentro de ella “data_dir”. En esta última dejaremos el archivo para publicar en la carpeta denominada “data”.

Shapefile

Para publicar un *shapefile*, primero hay que crear un espacio nuevo de trabajo en Geoserver, especificando el nombre del archivo y su URI (URL asociada a nuestro proyecto).

A continuación habrá que crear un Almacén de datos en la sección Datos – Almacenes de Datos.

- Al seleccionar “agregar almacén de datos” se debe definir el formato del archivo, en este caso será *shapefile*.
- También hay que escoger el espacio de trabajo definido anteriormente

Una vez obtenido el nuevo Almacén de Datos se creará una nueva capa escogiendo la anteriormente creada y seleccionaremos “publicación”.

- Hay que determinar el Sistema de Referencia de Coordenadas
- Se determinan también las dimensiones del encuadre, que se puede calcular desde los datos
- Por último, se debe determinar el estilo por defecto de la capa y clicar en “guardar”.

Tabla de BBDD espacial

El proceso para poder publicar una tabla de bases de datos espacial desde PostgreSQL, comparte varios pasos con la publicación del *shapefile*: el paso de la creación del “espacio de trabajo” es idéntico, por lo que no pasaremos a detallar nuevamente. El paso del “Almacén de Datos” es parecido, ya que se debe crear un Almacén de Datos, pero a la hora de elegir el formato, habrá que escoger “PostGIS-Database”.

Después se debe determinar en la ventana, el nombre de la base de datos, el esquema, el usuario y la contraseña, antes de guardar. Los siguientes pasos también son comunes a la publicación del *shapefile*, ya que se deben definir el SRC, las dimensiones y demás.

Una vez realizado estos pasos señalados, la tabla de BBDD espacial estará lista para poder publicarla.

8. Valoraciones y conclusiones

Para finalizar este documento se detallarán las conclusiones que se han obtenido tras implementar el sistema GIS para la recogida de datos en campo. Estas valoraciones se redactarán respecto a los requisitos propuestos por los técnicos del Departamento de Estadística Agraria y sensaciones propias tras finalizar la redacción del documento.

Los futuros usuarios de este sistema, los técnicos del Departamento propusieron dicho trabajo para poder mejorar la aplicación con la que trabajaban en el campo. Las mejoras que propusieron se han redactado en el apartado “Análisis y requisitos del proyecto” y en estas siguientes líneas se va a valorar el cumplimiento de todos ellos.

El primer requisito general que se ha planteado ha sido que el desarrollo de la aplicación debería ser en una plataforma de desarrollo abierto, aspecto que se ha completado ya que el software que se ha utilizado para la implementación de sistema GIS ha sido QGIS (sistema de código libre).

Una navegación eficaz por el visor fue otro requisito catalogado como no funcional, proporcionando una funcionalidad como la de ampliación/reducción de zoom. En este caso, este requisito se ha completado, ya que QGIS integra por defecto dicha funcionalidad. Por lo tanto, para acercarse a ciertos recintos o reducir zoom para tener una visión más amplia se podrá utilizar la rueda del ratón reduciendo el tiempo de espera que necesitaba el proceso en el anterior visor.

Relacionado con el anterior punto se encuentra también el siguiente. La actualización de la nueva vista tras la navegación por el visor se realiza ahora de una forma más ágil, reduciendo el tiempo de respuesta.

Un requisito indispensable era obtener una Base de Datos con la suficiente capacidad para albergar alrededor de un millón de registros correspondientes a los recintos de cada parcela de toda la Comunidad Foral de Navarra. La base de datos diseñada está capacitada para más registros de los que se han insertado en ella, por lo que los procesos que se le piden también se ejecutan con la suficiente rapidez y agilidad.

Además, el acceso a ella está solo accesible a los técnicos que serán los usuarios, ya que al abrir la base de datos se debe introducir una contraseña, haciendo dicho almacenamiento exclusivo para los usuarios del Departamento.

Por otro lado, los requisitos funcionales que propusieron los técnicos definían diferentes aspectos del visor del Sistema.

La visualización del mapa de Navarra se debe dividir en recintos de parcelas; esta información, se proporciona mediante la tabla “recintos” donde se encuentra el componente espacial. Cada registro en dicha tabla pertenece a un recinto, por lo tanto, este requisito también se ha completado.

La simbología también era punto a cumplir, ya que se debía de visualizar de la misma manera que se hacía en el antiguo visor. En éste, se pintaban los bordes de los recintos de un color diferente por cada grupo de cultivo. En el nuevo sistema se ha logrado la misma simbología como se ha podido observar en este documento.

Uno de los grandes inconvenientes del anterior visor, consistía en tener que abrir la base de datos desde el visor para editarla. En este caso, si se selecciona un recinto en el visor aparecen los atributos de dicho recinto en el formulario, acoplado en la parte inferior. Este funcionamiento permite una edición más rápida de la base de datos.

Asimismo, el editor montado en relaciones 1:N permite introducir diferentes apuntes para un mismo recinto, mejorando así, el anterior sistema. Éste solo permitía meter un cultivo por recinto (en el archivo DBF), por lo que se debía acceder a un formulario externo.

Otro problema que reducía la eficacia del trabajo de los técnicos en campo se encontraba en el acceso a las imágenes de base. El acceso a dichas imágenes requería desplegar un gran árbol de carpetas. En este caso, la ventana denominada “navegador” permite tener abiertas todas las carpetas y subcarpetas permitiendo acceder a ellas de forma inmediata.

A menudo los usuarios de este sistema trabajarán dividiendo el canvas del visor en varias vistas diferentes, para poder insertar en él diferentes mapas (NDVI, ortofotos de diferentes fechas etc.). QGIS ofrece para ello la posibilidad de insertar varias vistas de mapas, añadiendo al visor diferentes marcos con la posibilidad de añadir en cada uno de ellos, un nuevo mapa. Por lo que se puede concluir que este requisito también se ha completado.

Por último, era necesario proporcionar un mantenimiento anual sencillo para actualizar la información existente en la base de datos. Para ello, se les ha proporcionado a los técnicos las diferentes *queries* (consultas) que podrán ejecutarlas para poder actualizar la base de datos, haciendo así, un mantenimiento simple de la información.

En conclusión, este proyecto ha completado los requisitos que los técnicos propusieron al principio del mismo. Para cada requisito, se ha intentado buscar la solución más conveniente para el sistema y para el trabajo diario de los técnicos.

La estancia durante estos meses ha permitido la primera toma de contacto con el campo profesional, donde se ha podido integrar conceptos y procesos trabajados en el máster. En el aspecto personal, ha sido una experiencia muy enriquecedora trabajar al lado de profesionales del Departamento de Estadística Agraria del Gobierno de Navarra y agradezco en todo momento la atención de los técnicos para solventar las dudas que se han generado durante la realización del proyecto.

9. Bibliografía y webgrafía

Mención especial a los apuntes del Máster en Sistemas de Información Geográfica y Teledetección (MUSIGT). En especial, los vinculados a la asignatura Base de Datos Geográficas.

- Márquez, A. (2015). *PostGIS essentials*. Birmingham: PACKT publishing.
- Bassil, Y. (2012). A comparative study on the performance of the top DBMS systems. *Journal of Computer Science & Research*, 1.
- Elmasri, R., & B. Navathe, S. (2010). *Fundamentos de sistemas de bases de datos* (5th ed.) Pearson.
- Naiburg, Eric J. & Maksimchuk, Robert A. 2001. *UML For Database Design*. New York, NY: Addison-Wesley Publishing Company, Inc.
- Tudose, C., & Odubasteanu, C. (2010). *Database management systems comparative study: Performances of microsoft SQL server versus oracle*. (). Bucharest, Romania.

<https://www.qgis.org/es/site/about/index.html>

<https://products.office.com/es-es/access?rtc=1>

<https://www.postgresql.org/about/>

<https://dbeaver.io/about/>

<https://geoinnova.org/blog-territorio/ventajas-de-las-bases-de-datos-espaciales-frente-los-shapefiles/>

<https://www.intiasa.es/es/que-es-intia.html>

<https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales>

<https://postgis.net/>

<https://todopostgresql.com/extension-postgis-en-postgresql/>

<https://postgis.net/docs/manual-dev/postgis-es.html>

<https://www.tecnologias-informacion.com/basesdedatos.html>

<https://gis.stackexchange.com/questions/83918/best-choice-for-building-static-maps-postgis-spatialite-or-shapefile/83920>

Anexos

1- Comparación entre alternativas del proyecto

Comparación alternativas del proyecto

| | Compatibles estándares | Usuarios simultáneos | Funciones espaciales | Geometría por archivo | Triggers | N.º ficheros | Volumen datos | Importar/exportar datos |
|--|------------------------|----------------------|----------------------|-----------------------|----------|--------------|---------------|----------------------------|
| Postgis  | SI | SI | SI | Varias | SI | Varios | Amplio | Portabilidad baja(costosa) |
| Spatialite  | SI | NO | SI | Varias | SI | Único | Limitado | Portabilidad media |
| Geopackage  | SI | SI | Mediante ODBC | Varias | SI | Único | Amplio | Portabilidad alta |

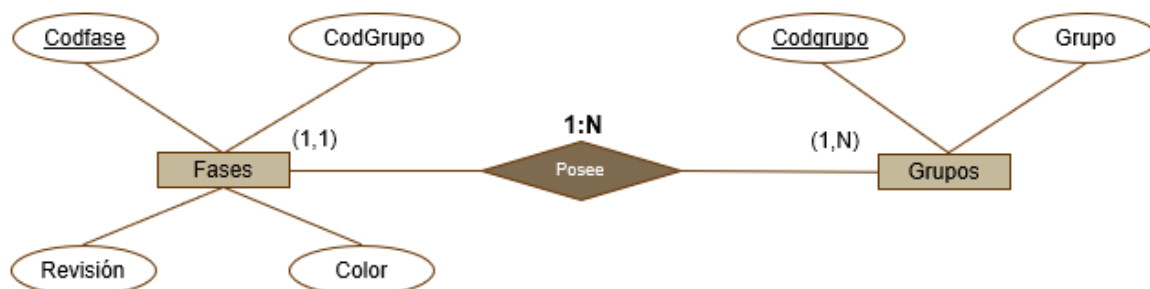
Limitaciones:

Postgis: su portabilidad es más costosa que en los otros formatos, ya que el número de ficheros es mayor.

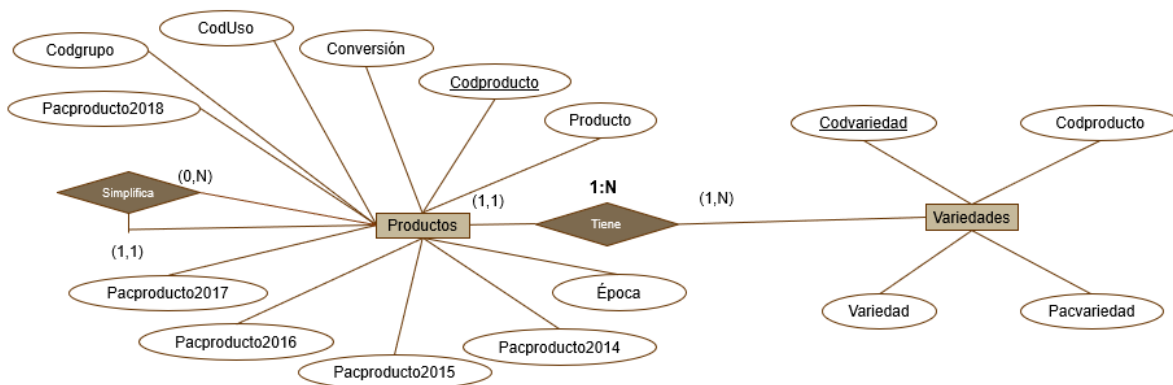
Spatialite: no permite que diferentes usuarios interactúen sobre el mismo archivo. Además, no es el formato más adecuado para grandes volúmenes de datos.

Geopackage: este formato está limitado por no incluir funciones espaciales, la solución es mediante el driver de ODBC. Además, no puede almacenar diferentes geometrías en un misma tabla.

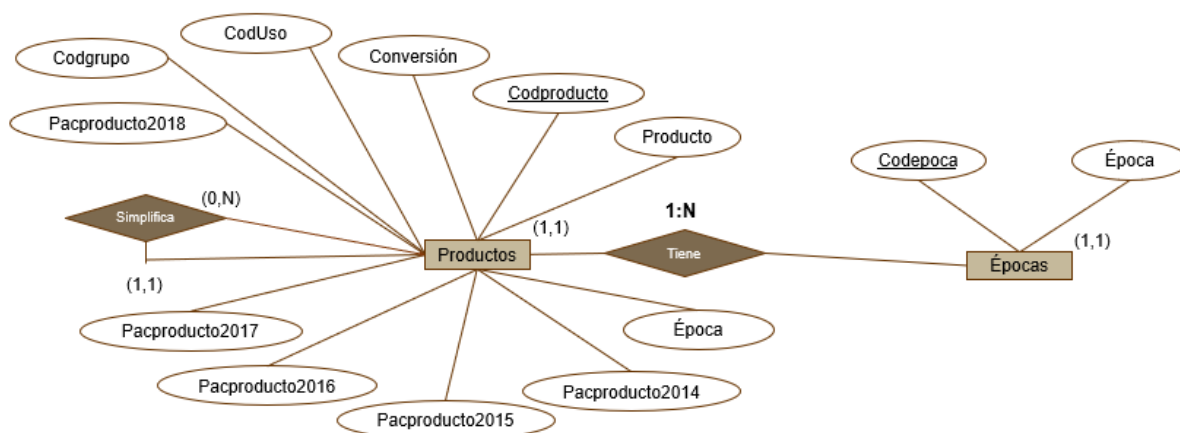
2- Relaciones entre tablas de “catálogo”



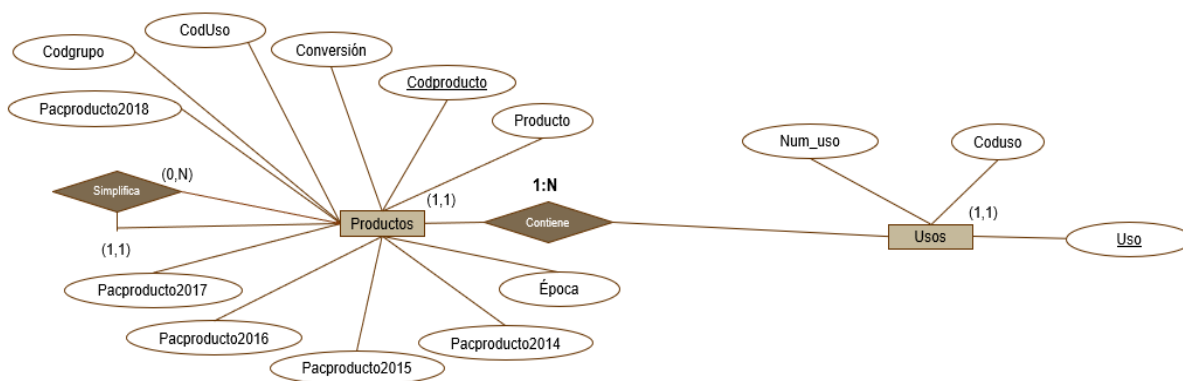
Relación Fases posee Grupos



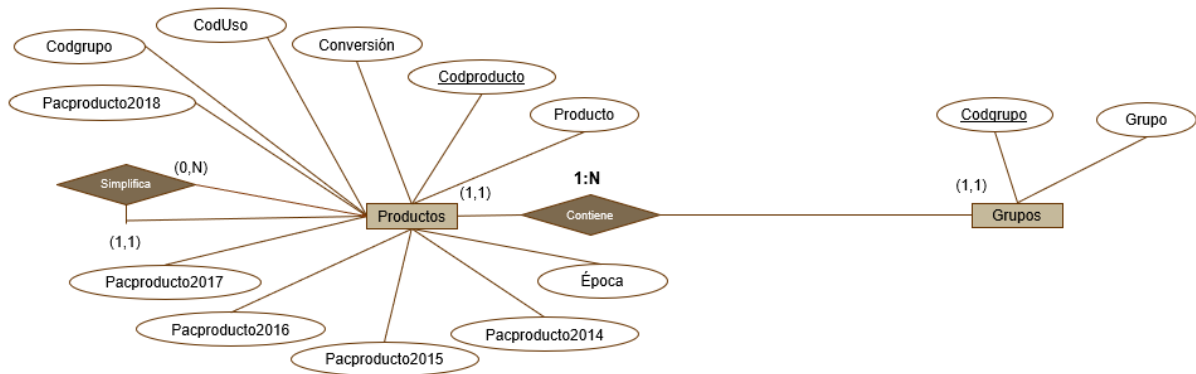
Relación *Productos* tiene *Variedades*



Relación *Productos* tiene *Épocas*



Relación *Productos* contiene *Usos*



Relación *Productos* contiene *Grupos*

3- Consultas de creación de “recintos” y “visitas”:

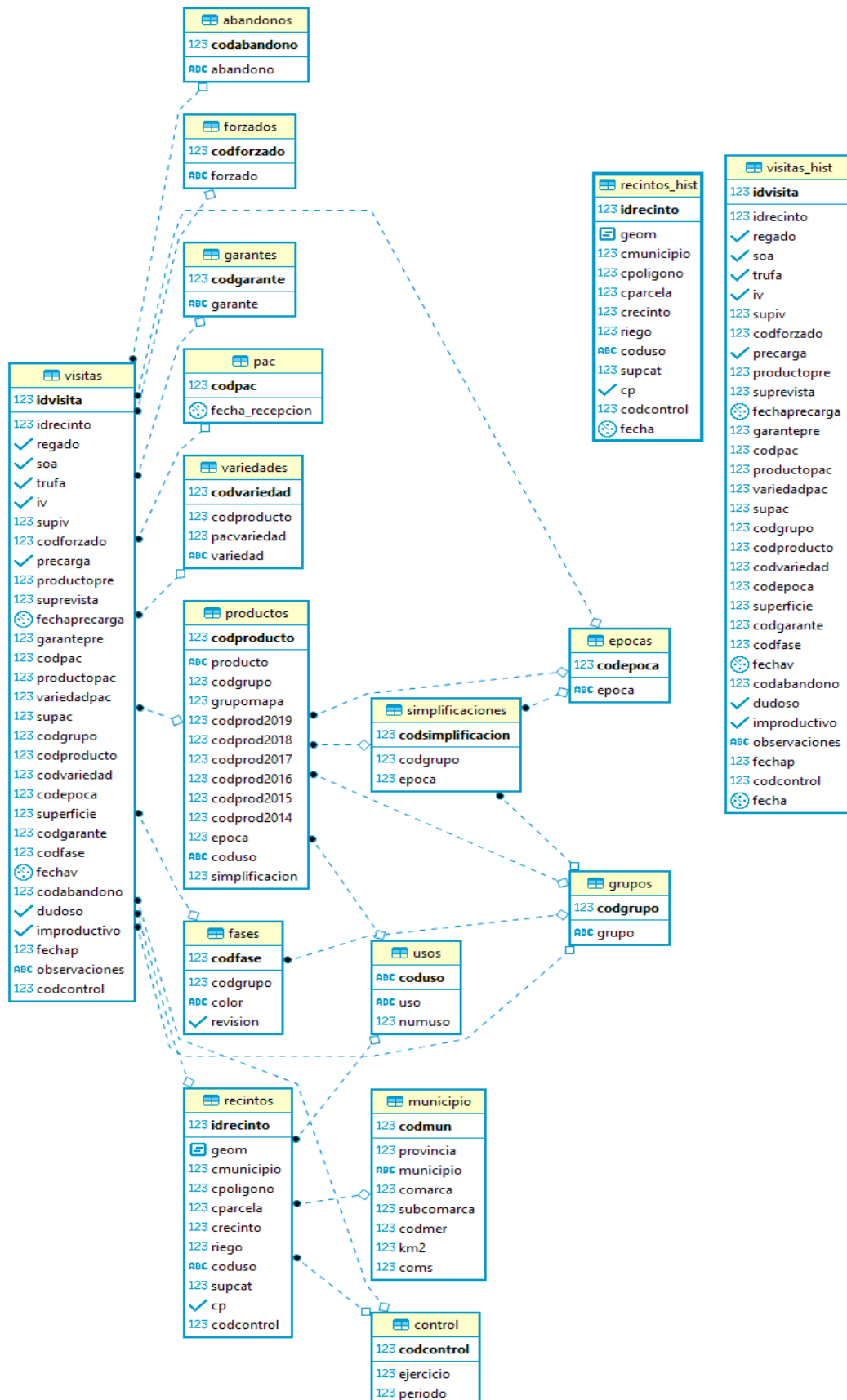
-Creación de la tabla “recintos”

```
CREATE TABLE proyecto.recintos (
  geom geometry NULL,
  cmunicipio int4 NULL,
  cpoligono int4 NULL,
  cparcela int4 NULL,
  crecinto int4 NULL,
  riego int4 NULL,
  coduso varchar(2) NULL,
  supcat float8 NULL,
  cp bool NULL,
  idrecinto int4 NOT NULL,
  codcontrol int2 NOT NULL,
  CONSTRAINT recintos_pk PRIMARY KEY (idrecinto),
  CONSTRAINT recintos_control_fk FOREIGN KEY (codcontrol) REFERENCES proyecto.control(codcontrol),
  CONSTRAINT recintos_fk FOREIGN KEY (cmunicipio) REFERENCES proyecto.municipios(codmun),
  CONSTRAINT recintos_uso_fk FOREIGN KEY (coduso) REFERENCES proyecto.usos(coduso)
);
CREATE UNIQUE INDEX recintos_cmunicipio_idx ON proyecto.recintos USING btree (cmunicipio, cpoligono, cparcela, crecinto);
CREATE INDEX sidx_recintos_geom ON proyecto.recintos USING gist (geom);
```

-Creación de la tabla “visitas”

```
CREATE TABLE proyecto.visitas (  
  idvisita int4 NOT NULL,  
  idrecinto int4 NULL,  
  regado bool NULL,  
  soa bool NULL,  
  trufa bool NULL,  
  iv bool NULL,  
  supiv float4 NULL,  
  codforzado int4 NULL,  
  precarga bool NULL,  
  productopre int4 NULL,  
  suprevista float4 NULL,  
  fechaprecarga timestamptz NULL,  
  garantepre int4 NULL,  
  codpac int4 NULL,  
  productopac int4 NULL,  
  variedadpac int4 NULL,  
  supac float4 NULL,  
  codgrupo int4 NULL,  
  codproducto int4 NULL,  
  codvariedad int4 NULL,  
  codepoca int4 NULL,  
  superficie float4 NULL,  
  codgarante int4 NULL,  
  codfase int4 NULL,  
  fechav timestamptz NULL,  
  codabandono int4 NULL,  
  dudoso bool NULL,  
  improductivo bool NULL,  
  fechap int2 NULL,  
  observaciones varchar NULL,  
  codcontrol int2 NULL,  
  CONSTRAINT visitas_pk PRIMARY KEY (idvisita),  
  CONSTRAINT visitas_control_fk FOREIGN KEY (codcontrol) REFERENCES proyecto.control(codcontrol),  
  CONSTRAINT visitas_epoca_fk_1 FOREIGN KEY (codepoca) REFERENCES proyecto.epocas(codepoca),  
  CONSTRAINT visitas_fase_fk_2 FOREIGN KEY (codfase) REFERENCES proyecto.fases(codfase),  
  
  CONSTRAINT visitas_fk FOREIGN KEY (codabandono) REFERENCES proyecto.abandonos(codabandono),  
  CONSTRAINT visitas_forzado_fk FOREIGN KEY (codforzado) REFERENCES proyecto.forzados(codforzado),  
  CONSTRAINT visitas_garante_fk FOREIGN KEY (codgarante) REFERENCES proyecto.garantes(codgarante),  
  CONSTRAINT visitas_grupo_fk FOREIGN KEY (codgrupo) REFERENCES proyecto.grupos(codgrupo),  
  CONSTRAINT visitas_pac_fk FOREIGN KEY (codpac) REFERENCES proyecto.pac(codpac),  
  CONSTRAINT visitas_producto_fk FOREIGN KEY (codproducto) REFERENCES proyecto.productos(codproducto),  
  CONSTRAINT visitas_recinto_fk FOREIGN KEY (idrecinto) REFERENCES proyecto.recintos(idrecinto),  
  CONSTRAINT visitas_variedad_fk FOREIGN KEY (codvariedad) REFERENCES proyecto.variedades(codvariedad)  
);
```

4- Diagrama del esquema "proyecto" con relaciones entre tablas



- 5- Ejemplo de un informe realizado por los técnicos: Informe de la semana del 5 al 12 de julio de 2019; catástrofe en Zona Media.

INFORME SEMANAL SOBRE SITUACIÓN DEL CAMPO

Semana nº 28 – Del 5 al 12 de julio de 2019

1- CLIMA Y SUELO

| | |
|--|-------------------|
| LLUVIA (Inferior a normal, Normal, Superior a normal) | SUPERIOR A NORMAL |
| TEMPERATURA (Inferior a normal, Normal, Superior a normal) | SUPERIOR A NORMAL |
| INFLUENCIA SOBRE LOS CULTIVOS (Beneficiosa, Nula, Perjudicial) (Ésta viene determinada por el clima (lluvia y temperatura)) | NULA-PERJUDICIAL |
| SUELO (Muy seco, Seco, Normal, Húmedo, Muy húmedo) | NORMAL-MUY HÚMEDO |

2. SITUACIÓN DE LOS CULTIVOS

| | Cosecha (1) | Situación (2) | Previsiones cosecha (3) |
|---------------------|-------------|---------------|-------------------------|
| CEREALES | N | 2 | -N |
| LEGUMINOSAS | N | 2 | -N |
| PATATA | N | 2 | -- |
| OLEAGINOSAS (Colza) | N | -2 | -N |
| HORTALIZAS | N | 2 | -N |
| FRUTALES | N-A | -2 | -N |
| VIÑEDO | N-A | 2 | -N |
| OLIVAR | N-A | 2 | N |
| PRADOS Y FORRAJES | N | 2 | +N |

(1) A: Adelantada, N: Normal, R: Retrasada.

(2) 1: Mala, -2: Inferior a normal, 2: Normal, +2: Superior a normal, 3: Buena.

(3) +N: Superior a normal, N: Normal, -N: Inferior a normal

3. SIEMBRA, PLANTACIÓN Y RECOLECCIÓN

| SIEMBRA Y PLANTACIÓN | Condiciones (1) | RECOLECCIÓN (2) | Rendimiento (3) |
|---|-----------------|---|-----------------|
| Maíz grano. Maíz forrajero. Maíz dulce. | B | Cebada. Avena. Trigo. Colza. Guisante seco. Haba seca. Ray-Grass. Alfalfa. Praderas. | N |
| Judía verde. Cardo. | B | Brócoli. Espinaca. Guisante verde. Escarola. Lechuga. Calabacín. | N |
| Invernadero: Lechuga. Ensaladas IV gama. Guindilla. Pimiento. | B | Invernadero: Lechuga, escarola, hortalizas baby leaf. Ensaladas IV gama. Tomate. Pimiento. Pocha. Alubia verde. | N |

(1) Condiciones en las que se está realizando la siembra y plantación: B: Buenas, R: Regulares, M: Malas. (2) Cultivos en recolección.

(3) Rendimiento esperado. I: Inferior, N: Normal, S: Superior, siendo N el rendimiento medio de la provincia.

4. DAÑOS

| ZONAS Y CULTIVOS AFECTADOS | AGENTE PRODUCTOR (Helada, sequía, inundación, incendio, ...) | CONSECUENCIAS (1) |
|--|---|-------------------|
| Varias zonas de Navarra (Cereales, frutales, hortalizas, viña) | Tormentas (granizo, lluvias torrenciales) | En valoración |

(1) L: Leve, G: Grave, MG: Muy Grave

Semana del 5 al 12 de julio de 2019

5.- OBSERVACIONES

Las tormentas descargan granizo y lluvias torrenciales en Navarra.

Climatología

El imprevisto que ha marcado negativamente la evolución de esta semana ha sido la llegada de una serie de tormentas el día 8 de julio, que causaron daños en diversos lugares, siendo el más lamentable la muerte de una persona arrastrada por las inundaciones. Estas tormentas descargaron granizo en algunas zonas y en otras produjeron unas lluvias muy intensas (más de 60 litros por metro cuadrado y hora) y persistentes (más de dos horas) que dieron lugar a fuertes riadas e inundaciones. Así, en la estación de Getadar se acumularon ese día 159,7 l/m², en la de Olite 87,9 l/m², en la de San Martín de Unx 83,7 l/m² y en la de Tafalla 100,2 l/m².

El Gobierno de Navarra celebró el día 11 de julio una sesión extraordinaria. La nota de prensa completa se puede consultar en el siguiente [enlace](#) y, a continuación, se transcribe la información de los acuerdos a los que se llegó en la misma que pueden ser de más interés para el sector agrícola:

- Se declara la situación de catástrofe, de acuerdo con la normativa foral, para, inicialmente, 14 municipios afectados el día 8 por las lluvias torrenciales registradas en la Zona Media (Barásoain, Beire, Ezprogui, Garínain, Leoz / Leotz, Lerga, Murillo el Cuende, Olite / Erriberri, Orísoain, Pitillas, Pueyo, Sada, San Martín de Unx y Tafalla).
- Se recomienda con carácter general a toda la ciudadanía, también al sector profesional del comercio, agricultura e industria, que presente reclamaciones por los daños ocasionados por la riada en sus propiedades y bienes ante las compañías de seguro. Es el primer paso, imprescindible, para poder acceder después a las compensaciones que correspondan o las que complementariamente puedan aprobar las instituciones públicas.
- Se habilitan aplicaciones específicas para que el sector agrícola y ganadero, con carácter informativo, pueda comunicar los daños registrados en sus [bienes afectos a las explotaciones agrícolas](#) e [infraestructuras de regadío](#). La Oficina Comarcal Agraria de Tafalla (AV Baja Navarra, 14) estará también abierta a diario en horario de 8 a 14:30 horas para prestar atención presencial al sector (teléfono 948703931).
- Se activarán aplicaciones específicas para que las entidades locales afectadas puedan comunicar los daños registrados en infraestructuras agrarias de carácter público y equipamiento público contemplado en el Plan de Inversiones Locales.

Por otro lado, se están evaluando los daños provocados por las granizadas registradas en otras tres zonas: en la Cuenca de Pamplona (principalmente en cereales); en la zona de Azagra, San Adrián y Sartaguda (sobre todo en frutales, horticolas y viña); y en la zona de Corella, Murchante y Tudela. Se espera que los daños sean altos en las zonas donde cayeron con más intensidad las tormentas y menores en el resto.

La estación de avisos de INTIA informa que ahora es el momento oportuno para realizar el control de determinadas plagas y enfermedades en diversos cultivos (se pueden consultar los detalles sobre éstos y otros temas fitosanitarios en la página <http://estacionavisos.agrointegra.intia.es/ai/accesovisor.do>).