

# Análisis del negocio de Prepersa para el diseño e implementación de un cuadro de mandos.

Guillermo Ruiz Guijarro

**Resumen**— Cualquier empresa actual dispone de una base de datos que se actualiza con nueva información prácticamente cada segundo. A menudo, el comportamiento de los datos que se están insertando puede determinar el funcionamiento de los proyectos relacionados. Sin embargo, extraer información útil de estos datos requiere de una inversión de trabajo que no todas las compañías quieren asumir. Este artículo trata sobre el proceso de análisis de la información de la empresa Prepersa para el posterior desarrollo de un cuadro de mandos que monitoriza datos de entrada y salida de la base de datos para la detección de situaciones desfavorables que puedan terminar ocasionando incidencias para el departamento de desarrollo de la empresa.

**Palabras clave**— Cuadro de mandos, MicroStrategy, entorno de trabajo.

**Abstract**— Any current company has a database that is updated with new information almost every second. Often, the behavior of the data being inserted can determine how related projects are working. However, extracting useful information from this data requires a work investment that not all companies want to take on. This article deals with the process of analyzing the information of the company Prepersa for the subsequent development of a dashboard that monitors data from the database's input and output for the detection of unfavourable situations that may end up causing incidents for the company's development department.

**Index Terms**— Dashboard, MicroStrategy, framework.



## 1 INTRODUCCIÓN

SEGUROS CATALANA OCCIDENTE [1] es una empresa que ofrece un conjunto de servicios de seguridad y cobertura de riesgos a sus clientes. Esta, forma parte de un grupo de aseguradoras que adopta este mismo nombre (Grupo Catalana Occidente), y que está formado, además, por las empresas Plus Ultra [2], Norte Hispa [3] y Seguros Bilbao [4]. Sin embargo, estas compañías no disponen de una plantilla de peritos puesto que estos, generalmente, trabajan de manera autónoma. Es aquí donde entra en juego la función de Prepersa [12], que se encarga de gestionar una red de peritos para cubrir las solicitudes que se les realizan a las distintas aseguradoras del grupo.

El departamento de desarrollo web de Prepersa trabaja en la creación de proyectos que responden a nuevas funcionalidades solicitadas, así como en el mantenimiento de aquellos cuyo desarrollo ya está finalizado y en funcionamiento. Es aquí cuando se presenta un escenario en el que los integrantes del departamento tienen que invertir parte de la jornada laboral en resolver incidencias relacionadas con el funcionamiento de los proyectos desarrollados y su uso. Es decir, actualmente existen situaciones en las que usuarios del portal tienen que hacer frente a problemas téc-

nicos que no están relacionados propiamente con su trabajo, siendo necesario desplegar una cadena de comunicación entre distintos departamentos para detectar que está existiendo un problema y así poder solucionarlo. Esto conlleva principalmente una problemática, la pérdida de tiempo de un profesional, que, a su vez, se traduce en la disminución de productividad del mismo y, en consecuencia, una pérdida de dinero a nivel empresarial.

El objetivo de este artículo es presentar una solución a este problema, utilizando la información disponible para la empresa y que ayudaría a detectar situaciones que pueden resultar en incidencias por parte de los usuarios. En el documento, por lo tanto, se explica el análisis que se ha realizado de los datos disponibles para Prepersa, así como el diseño y la implementación de un cuadro de comandos como solución a la problemática planteada.

## 2 ESTADO DEL ARTE

Durante el ciclo de vida de este proyecto, dos tecnologías han sido recurrentes, ya sea en la fase del análisis o en la de desarrollo, siendo estas Microsoft .NET por un lado y MicroStrategy por otro. En cualquier caso, estos nombres hacen referencia a frameworks utilizados para el desarrollo de aplicaciones y que están directamente relacionados con el trabajo realizado. El primero, es utilizado para la creación de aplicaciones web. El segundo, es utilizado para Business Intelligence (BI) y así obtener conocimiento del negocio a través de los datos generados.

- Email de contacto: guillermo.ruizg@e-campus.uab.cat
- Mención: TIC
- Tutor: Victor García Font (Departamento de Ingeniería de la Información i de las Comunicaciones. Área de Ciencias de la Computación e Inteligencia Artificial)
- Curso: 2019/20

## 2.1 Microsoft .NET

.NET es un framework desarrollado por Microsoft para el desarrollo de aplicaciones y que surge como respuesta para competir contra la plataforma Java de Oracle Corporation. Como cualquier framework, .NET está formado por un conjunto de herramientas con distintos objetivos, siendo ASP.NET una de ellas y que sirve para crear sitios web y aplicaciones mediante HTML, CSS y JavaScript. ASP.NET ofrece distintos marcos para crear aplicaciones, dando distintas alternativas para el desarrollo en función de distintos parámetros como los recursos, el tipo de aplicación al cual se quiere llegar o incluso la metodología a usar a la hora de trabajar.

.NET junto con ASP.NET permite la creación de Backends (BE) y Frontends (FE). Un BE es un código que se encarga de controlar la lógica, y, por lo tanto, el funcionamiento de la aplicación que se va a crear. En cambio, el FE será el encargado de organizar la información obtenida por el BE y disponerla visualmente mediante un entorno de trabajo concreto [5].

## 2.2 MicroStrategy

MicroStrategy es una compañía que ofrece un software que reúne distintos paquetes para el tratamiento de los datos, centrándose en la inteligencia de negocio e informes para empresas. La empresa se fundó en 1989 y actualmente tiene más de 2600 clientes, dando servicio a grandes entidades como por ejemplo el banco de Montreal [14]. La característica principal de este software es que, mientras otras alternativas se centran en ofrecer una de las opciones descritas a continuación, MicroStrategy ofrece todas las posibilidades bajo la misma arquitectura. Estos son los servicios que permite la herramienta:

**Informes empresariales:** Posibilidad de crear todo tipo de modelos que representan la información obtenida ya sea mediante informes o un cuadro de comandos.

**Análisis en cubo (MOLAP):** A partir de la predefinición de un conjunto de datos, la herramienta permite realizar distintas funciones para acceder a estos y tratarlos (hacer sumatorios, ordenarlos, etc.). Para ello, se necesita una base de datos multidimensional [13].

**Consultas Ad Hoc y análisis ROLAP:** ROLAP hace referencia a las bases de datos relacionales. MicroStrategy permite utilizar funciones como MOLAP y además permite al usuario crear sus propias consultas.

**Análisis estadísticos y minería de datos:** Se permite utilizar algoritmos para crear correlaciones entre los datos almacenados y patrones que se puedan estar dando.

**Envío de informes y alertas:** MicroStrategy ofrece la posibilidad tanto de crear alertas personalizadas como de tener un sistema centralizado de envío de las mismas.

## 3 OBJETIVOS

El objetivo principal de este proyecto es conseguir una aplicación para monitorizar información de tal manera que los integrantes del departamento de desarrollo web puedan tener, de manera rápida e intuitiva, una visión global del funcionamiento o estado de los proyectos en función

de los datos generados. Para ello, se deberán cumplir los siguientes objetivos (ordenados según su prioridad a la hora de ser realizados):

- Analizar el funcionamiento de los distintos módulos a monitorizar para detectar el modelo de datos que existe detrás de los proyectos.
- Desarrollar un producto mínimo viable, que ofrezca una funcionalidad completa y que satisfaga las expectativas de los interesados.
- Diseñar la aplicación de tal manera que se puedan detectar situaciones precedentes a un caso de incidencia y así lograr anteponerse a ella antes de que suceda.
- Conseguir un producto robusto, que permita una fácil adaptabilidad en el caso de futuras peticiones y sea óptimo en cuanto a tiempos de ejecución.
- Disponer de valores históricos para poder hacer comparativas del funcionamiento de un cierto elemento en función del tiempo.

## 4 METODOLOGÍA

### 4.1 Scrum

La técnica elegida mediante la que se pretende desarrollar el proyecto es Scrum. Esta, es una metodología ágil que se utiliza para el desarrollo de software cuya característica principal es la iteración. Esta forma de desarrollar proyectos se basa en un conjunto de principios entre los que se destacan: Satisfacer al cliente mediante la entrega temprana y continua de software con valor; aceptar cambios de requisitos en cualquier etapa; trabajo cotidiano entre responsables del negocio y desarrolladores; facilita la comunicación entre el equipo e intenta reflexionar sobre cómo mejorar para ser efectivos en intervalos regulares [6]. En concreto, Scrum aplica estos conceptos utilizando un conjunto de elementos formado por el equipo (con sus respectivos roles), bloques de tiempo, artefactos y reglas concretas.

En el caso de la empresa donde se desarrolla este proyecto, los roles existentes son el de **Scrum Master** (responsable del equipo y figura encargada de velar por la integridad de la metodología) y el **equipo** (encargados de desarrollar las distintas funcionalidades del sistema). En este caso, el Scrum master se encarga de gestionar el **Backlog** (conjunto de tareas que se desarrollaran en alguna iteración del proceso), responsabilidad que realmente recae en el **ProductOwner** (responsable de maximizar el valor del producto con el objetivo de analizarlo desde el punto de vista del negocio).

Los bloques de tiempo que definen la metodología son las **Sprint Plannings** (se escogen las tareas del Backlog que se van a desarrollar durante el sprint); los **Daily Meetings** (reuniones cortas en las que cada miembro del equipo comunica el estado de su trabajo y plantea dudas o resoluciones); el **Sprint** (tiempo que se dedica a trabajar en las tareas seleccionadas); la **Review** (reunión para analizar cómo ha ido el Sprint y que es aquello que se puede mejorar o incluso adaptar del proyecto) y la **Retrospectiva** (se incorpora el feedback obtenido en la Review y se analizan todos

los aspectos del Sprint realizado).

Los artefactos usados en esta metodología se centralizan todos en la aplicación JIRA. En ella, podemos encontrar el **Product Backlog** (tareas pendientes que debe realizar el equipo); el **Sprint Backlog** (tareas que el equipo debe cumplir durante la duración del Sprint) y la **BurndownChart** (gráfica que determina el trabajo realizado y las horas pendientes que le faltan al equipo antes de finalizar el Sprint) [7].

En Prepersa, una iteración dura dos semanas. El lunes de la semana 1, se realiza una planificación de las tareas que se van a realizar durante el sprint. El martes, se realiza la retrospectiva del sprint pasado y el resto de la semana se prosigue con el sprint. La siguiente reunión (Review) se realiza el viernes de la semana 2. Diariamente se hacen los daily meetings, teniendo estos una duración aproximada de 15 minutos.

El motivo por el que se ha escogido esta metodología es que, aunque el trabajo se haya asignado para ser realizado de manera individual, está totalmente rodeado del negocio (con todo lo que eso conlleva) y será necesario cooperar con los compañeros del departamento para desarrollar el proyecto. Se ha decidido que el autor de este trabajo también asista a las reuniones para poner en común el progreso y que así todos los miembros del equipo puedan aportar ideas o ceder ayuda y/o conocimientos, sobretodo más propio del negocio, para analizar de manera correcta la información obtenida. Por lo tanto, aunque al trabajar de manera individual pueda parecer que no tenga sentido usar Scrum, realmente se considera este proyecto uno más del departamento, y, por lo tanto, se cree oportuno que se proyecte el avance en el equipo. Además, con esto se pretende ver como el software va avanzando de manera iterativa, sin tener que esperar al final del proyecto un resultado definitivo, sin tiempo de reaccionar ante posibles errores.

## 4.2 Test de usabilidad

Otra metodología que se pone en práctica en una parte del ciclo de vida del proyecto son los test de usabilidad, definidos por Dumas & Redish [8] y que sirven para detectar si la interfaz de usuario funciona tal y como se esperarían todo aquel que la vaya a utilizar.

Dumas & Redish defienden que, para la realización de un test completo, se deben seguir los pasos mencionados a continuación. En primer lugar, se debe planificar el test, contestando a las preguntas de cuándo, cómo, dónde, quién, por qué y qué, así como el propósito del mismo y los objetivos. El siguiente punto consiste en seleccionar los participantes que realizarán el test, escogiendo personas que representen distintas características. Otro paso a contemplar es la preparación del material, incluyendo el prototipo o aplicación, guión de la sesión, autorización de cesión de datos, etc. En el momento del test, es importante remarcar al usuario que verbalice todo lo que hace, para entender el por qué de los errores y detectar cual es la visión que tiene una persona externa totalmente desvinculada del diseño de la aplicación. Por lo tanto, es importante hacer un seguimiento de la sesión con preguntas, que, sin ser invasivas, inviten al participante a comentar todo lo que está haciendo. Finalmente, el último paso a realizar

consiste en formalizar la información obtenida con la sesión para poderla comparar con la de otras y así obtener unos resultados que definan los cambios y aspectos a mejorar de la interfaz [9].

## 5 ELICITACIÓN

El primer paso para la realización del proyecto consiste en la elicitación o toma de requerimientos, para entender que es exactamente lo que buscan los clientes y lograr tener una visión común del producto final al que se quiere llegar.

En cuanto a estos requisitos, se pueden definir como las características que debe cumplir un sistema para satisfacer una necesidad concreta. Estos, se dividen en dos tipos, los funcionales y los no funcionales. Los primeros, tal y como dice su nombre, definen una funcionalidad, es decir, una reacción que tendrá la aplicación a unas entradas y que provocará unas salidas. En cambio, los no funcionales son aquellos que definen una característica sobre cómo será algo, ya sea en términos de rendimiento o manera en que se muestre la información, entre otros [10].

Para este proyecto, se ha recurrido al uso de entrevistas con los interesados en el proyecto del departamento para obtener los requisitos de la aplicación, puesto que, aunque usar más técnicas puede ayudar a ahondar en la definición del producto, realizar una elicitación extensa no es uno de los objetivos que se quieren realizar en este proyecto y que formaría parte de un trabajo de la rama de la ingeniería del software. Aún así, se han llevado a cabo las prácticas recomendadas a la hora de realizar una entrevista para obtener información útil y de calidad.

A continuación, se muestran los requerimientos más destacables para la aplicación ya que son los que definen su principal funcionamiento y dan una línea base que responde a cómo será el cuadro de comandos.

Funcionales:

- Cuando el usuario accede al dashboard, este debe mostrar 5 gráficos referentes a los módulos de la apertura de encargos, cierres de encargos, mensajes, pagos y umbrales.
- Si un usuario hace clic en el título de un gráfico, se navegará a una pantalla con información detallada del módulo.
- Para los módulos de mensajes y cierre de encargos, la aplicación deberá mostrar un listado con los registros que han fallado.
- Para los módulos de aperturas de encargos y pagos, la aplicación deberá mostrar una alerta si no ha entrado ningún registro en un cierto periodo de tiempo.
- El sistema debe permitir comparar los valores obtenidos con otros históricos.
- Para los cierres de encargos, se dividirá la información por grupo profesional.

No funcionales:

- Cuando la aplicación muestre datos de rendimiento en cuanto a tiempo, la unidad será en segundos.
- La aplicación no deberá tardar en cargar una pantalla más de 10 segundos.

## 6 ANÁLISIS

El propósito de este apartado es el de mostrar todo el proceso de análisis que se ha seguido y cuyo objetivo es el de definir como va a ser la aplicación, como va a estar estructurada y que información se va a mostrar en ella.

La primera fase del análisis consiste en entender que es aquello que se quiere monitorizar. Antes de modelar una primera versión de la aplicación, es primordial conocer qué datos se deberán mostrar, para así poder generar luego una estructura que organice la información. En este proceso se considera necesario, en primer lugar, analizar las incidencias que recibe el departamento, para así encontrar patrones o errores similares que puedan apuntar a algún motivo más general. Es aquí dónde se detecta que la gran parte de las problemáticas que tiene que resolver el departamento están estrechamente relacionadas con todo lo referente a los encargos. Un encargo es una solicitud que se transmite de un cliente a un perito. Se puede entender como un pedido a realizar o problema a solucionar. Para ejemplificar esto, un perito especializado en automóviles puede tener como encargo asignado inspeccionar un vehículo que ha sufrido un siniestro.

Viendo esto, se decide que el estudio se debería hacer entorno a los mismos, puesto que se podría cubrir un amplio abanico de problemáticas dónde existe algún tipo de dependencia con los encargos.

### 6.1 Módulos

Tras exponer esta idea a los interesados del departamento, se trabaja en el estudio de los módulos que conformaran la aplicación. Estos, surgen después de celebrar una reunión donde distintos miembros del equipo, entre ellos representantes del departamento de calidad, exponen ideas para esta primera versión del aplicativo. El resultado de dicha reunión se materializa en los siguientes 5 módulos expuestos a continuación.

- Apertura de encargos. Módulo para mantener un control de la información que se genera de los nuevos encargos creados. Se realiza un seguimiento del estado de los mismos.
- Cierre de encargos. Control del cierre de los encargos. Este cierre depende de un servicio externo no controlado por el departamento, y, por lo tanto, es necesario conocer el estado de las entradas y salidas de este servicio para detectar errores.
- Pagos realizados. Junto al cierre, un encargo debe ser debidamente remunerado. Mantener un control de los pagos es necesario para confirmar que el ciclo de vida del encargo se está cumpliendo.
- Seguimiento de mensajes. El sistema que utiliza Prepersa para comunicarse con las compañías a las que distribuye peritos genera mensajes que son almacenados en los sistemas de la empresa. Es necesario mantener un seguimiento del estado de los mismos para saber si la comunicación entre compañías se está llevando a cabo correctamente, y, por lo tanto, se están resolviendo los encargos debidamente.
- Umbrales de los profesionales. Cada profesional

tiene asignado un límite de encargos a realizar. Se quiere mantener un control de dicho límite puesto que actualmente se están generando datos inconsistentes y es necesario realizar modificaciones manuales. Por lo tanto, es necesario monitorizar esta situación para modificar los datos de la manera más rápida posible.

### 6.2 Procedimiento

Esta subsección explica el procedimiento que se ha seguido para realizar el análisis de los módulos previamente mencionados.

El primer paso consiste en acceder al portal web para detectar la pantalla del módulo que se quiere empezar a analizar. Una vez detectada, se realiza un estudio de los datos gestionados en dicha pantalla. En ocasiones, estas muestran información y en otras es demandada para que sea insertada por el usuario y poder ser guardada en la base de datos. En cualquier caso, al detectar la pantalla que nos interesa, se comprueba su código HTML para ver cual es el proyecto de FE que existe detrás y que se encarga de mostrar los datos.

Por lo tanto, conociendo este FE, podemos acceder a su código para detectar los BE asociados, y que son los que crean, modifican o eliminan la información mostrada.

Al conocer el BE en cuestión, se puede proceder a realizar un análisis del código para ver como se está manipulando la información, que tablas de la base de datos se están utilizando, en cuales se insertan información, dónde se realizan actualizaciones de los datos y el motivo, etc. Conociendo estos aspectos, se consigue entender el funcionamiento del módulo, así como el origen y el destino de los datos, lo que permite que se pueda empezar a pensar en cómo monitorizarlo.

Cabe destacar en este punto que, tanto los FE como los BE, tienen unas entradas y unas salidas con las que se define el flujo de los datos. Esto significa que, en función del valor de las salidas de un BE, la pantalla de FE podrá ser distinta, lo que conlleva a que parte del análisis consista en saber discernir entre los valores que la salida puede tomar y cuáles son aquellos que hacen que se acceda a la pantalla deseada.

Una vez realizado este seguimiento desde la pantalla principal hasta el BE, ya conocemos el flujo de la información y su origen, lo que nos permite definir cuáles serán las gráficas a realizar para monitorizar la información.

En la figura 1 se puede observar la estructura conformada por los distintos elementos mencionados anteriormente. Se muestra un ejemplo de una situación en la que, un BE, en función de algún valor de la salida, accederá a un FE u otro. Tal y como se ha mencionado anteriormente, es importante entender cual es el flujo de la información, puesto que un simple detalle puede provocar la navegación a pantallas distintas.

Fruto del análisis realizado, se obtiene la estructura del modelo de datos utilizado en la aplicación. En la figura 2 se pueden observar las tablas que contienen algún tipo de información que será monitorizada o de la que se derivarán datos para mantener un seguimiento, así como la cardinalidad entre ellas.



- MicroStrategy ya controla el acceso a los proyectos por parte de los usuarios sin la necesidad de tener que diseñar un sistema de registro y log-in.

Tras obtener estas conclusiones después de realizar la comparativa, se decide finalmente utilizar MicroStrategy como framework de desarrollo de la aplicación. La herramienta, ofrece la profundidad suficiente como para obtener el resultado esperado. Además, incluir nuevas librerías de JS para que sean soportadas por la arquitectura de la empresa conlleva el planteamiento de aspectos como la seguridad, y no se considera que esta opción aporte beneficios suficientes como para considerarla por encima de MicroStrategy.

## 7 IMPLEMENTACIÓN

Este apartado del artículo se centra en explicar cual ha sido el proceso de desarrollo e implementación de la aplicación. Una vez seleccionada la herramienta de desarrollo y conocido el producto a crear, se procede a implementar la aplicación. De manera general, esta fase del proyecto se divide en tres etapas: la creación de consultas SQL, la creación de informes mediante MicroStrategy Developer y finalmente la representación gráfica de los mismos mediante MicroStrategy Web.

Con la primera herramienta, crearemos los informes, mediante los cuales se recopilará la información. Esta, es la que nos permitirá crear el BE de la aplicación, obteniendo todos los datos de la base de datos de Prepersa. Con la segunda, usaremos los informes creados para generar los distintos gráficos que usaremos en la aplicación. Por lo tanto, es con esta con la que generaremos el FE del aplicativo. De esta manera, existirá una dependencia en el desarrollo ya que la parte visual necesitará los reportes generados. Con esto se pretende facilitar el mantenimiento del dashboard en un futuro, independizando la obtención de datos de su representación. Gracias a ello, lo que conseguimos es que, si se necesita crear una nueva representación de la información de un conjunto de datos que ya traemos de la base de datos, directamente con el acceso web podremos crear la nueva vista. Sin embargo, si, por ejemplo, un gráfico de barras muestra el valor de los últimos 5 días trabajados y se quiere ampliar el intervalo de días, la vista no se modificará y desde la herramienta Developer se trabajará para modificar las condiciones que sean necesarias.

Cabe destacar que desde la aplicación web también se podría acceder directamente a la base de datos, pero tener dos orígenes de datos implicaría una mayor dificultad a la hora de corregir errores. Además, se considera apropiado separar la funcionalidad del BE con la del FE, separando así las acciones posibles a realizar entre las dos herramientas.

Por lo tanto, el proceso de desarrollo consistirá en crear los informes, obteniendo los datos de nuestra base de datos de Prepersa para seguidamente generar los gráficos que consumirá el usuario.

En la figura 4 se muestra la arquitectura de la aplicación, estando esta dividida en BE y FE. Como se puede observar, el usuario accede a los gráficos que estarán informados con los datos de los informes obtenidos de la base de datos.

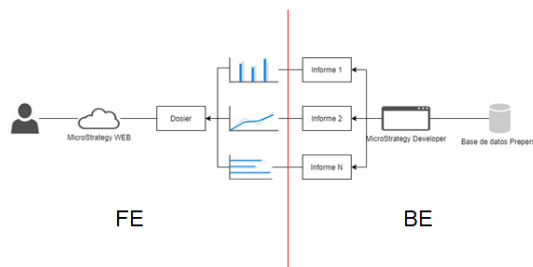


Fig. 4: Arquitectura de la aplicación.

### 7.1 Creación de consultas SQL

El primer paso a la hora de realizar la implementación del producto es la creación de las consultas SQL para obtener la información. Estas, no se usan directamente en MicroStrategy, puesto que el framework es capaz de generarlas automáticamente. Sin embargo, las realiza a raíz de la creación de unos objetos y la relación establecida entre ellos. Lo que esto significa es que, todo y no ser un trabajo primordial, se ha considerado necesario realizar todo el modelado de las consultas para comprobar los resultados obtenidos y tener así una base sobre la que comprobar los datos conseguidos mediante el uso del framework.

### 7.2 Creación de informes

La creación de informes es la parte del desarrollo en la que se extrae la información de la base de datos. Gran parte del peso de la implementación del cuadro de comandos recae en este momento, puesto que se basa en materializar todos esos datos a monitorizar fruto del análisis realizado previamente.

El primer paso en esta fase consiste en la configuración de las tablas. MicroStrategy permite la importación de tablas de una base de datos, pero no mantiene ningún tipo de relación entre ellas. El problema principal que esto conlleva es, por lo tanto, que la relación que se hace por defecto entre dos tablas A y B consiste en multiplicar las N filas de A por las M de B, generando un informe C con  $N \times M$  registros (producto cartesiano), un resultado que no aporta información útil. Para evitar esto, hay que indicar la cardinalidad entre las distintas tablas. Como en la fase de análisis se obtuvo el modelo de datos a utilizar, ahora el trabajo consiste en plasmar dicho modelo con MicroStrategy. Para ello, se ha creado un atributo para cada una de las 20 tablas que conforman el modelo. Un atributo representa un valor que da información contextual al dato, como por ejemplo un nombre o una fecha. Por lo tanto, mediante este tipo de objeto se han relacionado las tablas entre ellas para obtener el modelo de datos mostrado en la figura 2.

Con esta configuración, es el momento de generar los informes. Cada una de las gráficas que conformarán el cuadro de comandos tendrá un informe asociado, siendo por lo tanto necesario crear un total de 33 informes.

La creación de un informe consiste en la correcta combinación de objetos para la obtención de la consulta deseada en base de datos. Para ello, se han utilizado, además de los atributos, los hechos (valores numéricos que

dan alguna información del negocio), métricas (cálculos realizados a raíz de los hechos), filtros (condiciones que especifican el rango de datos) y grupos especializados (agrupaciones de datos en función de un criterio).

Para cada uno de los informes, por lo tanto, se ha seguido el siguiente proceso. En primer lugar, se detectan los atributos que se deban generar y que contextualicen la información, así como por ejemplo lo puede ser una fecha o el estado de un mensaje.

Una vez se tienen los atributos creados (haciendo referencia a las columnas de la tabla de la base de datos necesarias) se crean los hechos para obtener los datos numéricos, como puede ser, por ejemplo, el número de encargos pendientes que tiene un profesional asignados.

El siguiente paso consiste en generar las métricas necesarias para realizar el cálculo pertinente con la información obtenida de los hechos. Hay que analizar que es aquello que se quiere representar y utilizar la fórmula o función que proporcione el dato concreto. En este caso, el cálculo puede ser algo sencillo como una agregación (count, sum, etc.) o más elaborados como estadísticas para comprobar cómo se están comportando los datos en función del tiempo.

Finalmente, se añaden los filtros, mediante los cuales se evalúan los atributos para obtener un conjunto de datos bajo una o más condiciones. Por ejemplo, en el proyecto se utilizan filtros para obtener información de las últimas 24 horas o de los encargos que se encuentran en un cierto estado.

En la figura 5 se muestra la relación entre los elementos del proyecto.

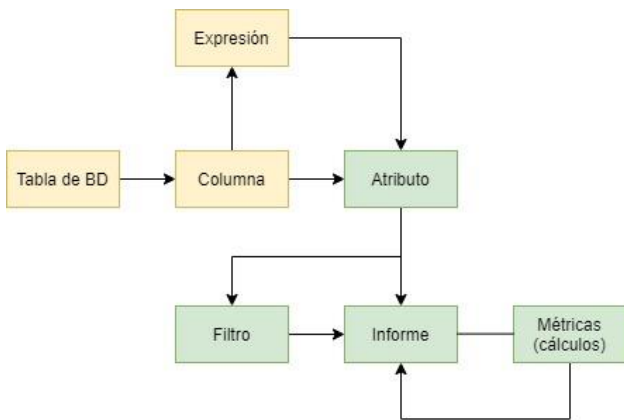


Fig. 5: Relación entre los elementos del proyecto para la creación del informe.

### 7.3 Creación de gráficos

Con los informes creados, ya disponemos de los datos necesarios para crear los gráficos. Para ello, nos ayudaremos de MicroStrategy Web, herramienta que nos permite crear las representaciones gráficas de la información.

Por lo tanto, el primer paso que se lleva a cabo es la creación del dossier, un objeto que agrupará todos los elementos del proyecto creados anteriormente. Este, se divide en capítulos, y a su vez, en páginas. Esta organización nos permite crear un capítulo para cada módulo a representar

y tantas páginas como pantallas necesitemos para dicho capítulo (o módulo).

Una vez creada la estructura del dossier, el siguiente paso será realizar la importación de la información. En este caso, como los informes ya han sido generados en el apartado anterior, indicaremos a la herramienta que los queremos usar como fuente de datos.

Ahora, el proceso seguido se basa en añadir tantas ventanas de visualización como distintos gráficos queramos tener en una misma pantalla, e indicar el método de representación de la información y el informe del cual se obtendrán los datos. Como los atributos y los cálculos pertinentes ya han sido creados en el proceso de generación de informes, ahora simplemente se tiene que indicar cuál se quiere utilizar en la vista en la que nos encontremos.

Por lo tanto, imaginemos ahora que se quiere crear la pantalla de mensajes. En primer lugar, se deberá generar un capítulo del dossier con las páginas que contendrán los gráficos.

Una vez creado el espacio donde se dispondrán las visualizaciones, se crean las distintas ventanas y disponen en función de cómo se desea que se muestre la información (Figura 6).

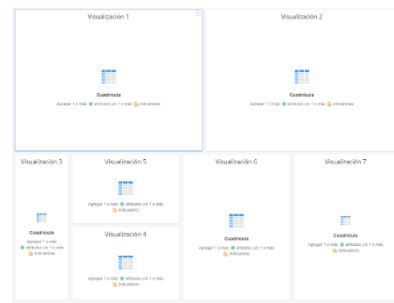


Fig. 6: Estructura de las visualizaciones de la pantalla.

Una vez creada la estructura, se debe indicar la fuente de datos para la visualización. Para ello, se incluye un informe generado anteriormente con MicroStrategy Developer.

Cuando se tiene el informe agregado al proyecto, se determina el tipo de gráfico que se usará en la visualización.

Finalmente, se indica en el menú contextual los atributos y métricas que se quieran mostrar. Aquí, también se indica si se quiere que la información esté agrupada de alguna manera en específico. Por ejemplo, tal y como se ha comentado en el apartado 7.2, hay un tipo de objeto llamado grupo personalizado que sirve para juntar los datos en función de una característica. Por lo tanto, si se busca dividir los mensajes en función de si comunican una inspección o un siniestro, se usará un grupo personalizado para indicar a la herramienta que genere el *Group By* correspondiente.

Siguiendo esta metodología se puede ir configurando la pantalla para acabar obteniendo el resultado esperado (Figura 7).

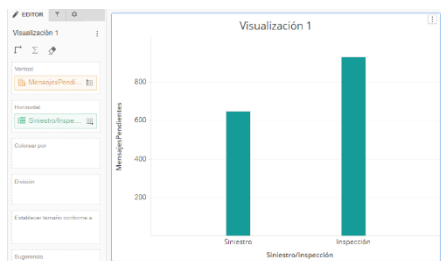


Fig. 7: Visualización generada con métricas y atributos.

## 8 PRUEBAS

Esta sección explica como han sido realizadas las pruebas sobre la aplicación. En primer lugar, cabe destacar que se han hecho dos tipos distintos de pruebas, haciendo énfasis en los dos aspectos principales para que la aplicación sea funcional: los tiempos de carga y la interfaz de usuario.

### 8.1 Tiempos de carga

Para comprobar que los tiempos de carga son aceptables, se ha procedido a realizar ejecuciones manuales del dashboard en diferentes franjas horarias, para realizar la carga de datos en distintos contextos, donde la base de datos tenga cargas de trabajo diferentes. Por lo tanto, el procedimiento seguido ha consistido en realizar ejecuciones a las 9:00 (inicio de la jornada laboral), 12:00 y a las 16:00. Se ha medido el tiempo entre la realización de la petición (figura 9) de acceso al dashboard hasta su total carga de los datos en las tres franjas horarias indicadas. Estas, han sido elegidas de tal manera para representar distintos estados de la base de datos. A las 9:00, comienza la jornada de trabajo y es una franja donde generalmente, no hay sobrecarga de la base de datos. Tanto el acceso por parte de los departamentos de la oficina como de los peritos es menor. Luego, la franja de las 12:00 es una donde la base de datos está más concurrida. Se ha seleccionado esta hora porque es posterior al tiempo de descanso, pero anterior a la franja permitida para comer, por lo tanto, se puede considerar una hora con niveles altos de trabajo. Es aquí por lo tanto donde se ha querido realizar otra ejecución del dashboard. Finalmente, las 16:00 es una hora donde, generalmente, la productividad no es tan alta, pero superior a la primera franja de testeo, por lo que de esta manera se pueden obtener valores con tres estados distintos de la base de datos.

Con esta prueba, realizada durante una semana, se han obtenido los valores referentes a los tiempos de carga del dashboard en segundos (s) de 9,14 s para las 9:00; 11,02 s para las 12:00 y 9,73 s para las 16:00. Los resultados han sido aceptables, sin embargo, en el apartado 9 se hace una valoración de los mismos y se propone una solución para mejorarlos.

### 8.2 Pruebas de interfaz

Para realizar las pruebas referentes a la interfaz, se ha hecho uso de la metodología explicada en el apartado 4.2 del artículo. Se ha desarrollado un guión (sección Apéndices A.1) donde de manera pautada y concisa se solicita al

colaborador que está realizando el test que haga unas acciones determinadas con el dashboard para valorar como de intuitivo es. Gracias a esta prueba se determina, por un lado, si la información está correctamente organizada y por lo tanto se puede navegar por el cuadro de comandos tal y como se tiene previsto. Por otro lado, se comprueba si la información que se ha decidido utilizar permite lograr el objetivo de intuir situaciones que puedan desencadenar algún tipo de incidencia.

Los resultados de la prueba indican que la pantalla principal no cumple con el objetivo de ser intuitiva. Para los implicados en la prueba, la lectura de los datos del cierre de encargos es difícil de entender. El motivo principal es que estos están divididos en solicitudes realizadas, pendientes y cerradas, con un eje distinto para cada uno de los grupos, y escalas distintas. Esto, ha producido malinterpretaciones por parte de los realizadores de la prueba. Para los casos de la apertura de encargos y pagos realizados, la problemática ha sido similar. Se comparan valores en ejes distintos, lo que ha supuesto un esfuerzo más elevado del esperado en la interpretación de los datos. Para el caso de los umbrales, donde la información está dispuesta en forma de tabla, se ha considerado que los datos mostrados no permiten determinar si el módulo está funcionando correctamente o no. En el apartado de resultados del artículo se explica como proceder para solventar esta problemática. El resto de las pantallas muestran la información, generalmente, de manera intuitiva y fácilmente interpretable.

## 9 RESULTADOS

La propuesta de crear un cuadro de comandos para la monitorización de datos ha sido completada de manera satisfactoria. Mediante MicroStrategy, se han generado los gráficos que devuelven los datos obtenidos de la base de datos de manera correcta. Haciendo referencia a los objetivos propuestos en el apartado 3 del documento, a continuación, se detallan los resultados obtenidos en relación a los mismos.

- Se ha conseguido realizar un análisis de los distintos módulos de manera efectiva. Esto, se puede demostrar puesto que se ha detectado el concepto de encargo como problemática principal y motivo de las incidencias y se ha hecho un análisis alrededor de este núcleo. Prueba de ello, es que se ha podido representar el modelo de datos que hay detrás de esto, comprendiendo la relación entre los elementos y extrayendo información útil de los mismos.
- Una vez mostrado el prototipo, los interesados decidieron que sería un buen resultado obtener el desarrollo de dicho ejemplo la cual cosa se ha conseguido completar. El cuadro de comandos es funcional y se representan los módulos propuestos en la fase de prototipaje. Sin embargo, cabe destacar que se ha buscado representar de manera visual un resultado binario del estado de cada módulo (existe o no existe error) a parte de la propia información representada. Esto, no se ha podido representar mediante la herramienta utilizada, pero se

- exploran maneras para conseguirlo en un futuro.
- Los datos de la aplicación se actualizan cada 30 segundos, por lo tanto, permite detectar situaciones que se puedan considerar sospechosas. Este cuadro de mandos no determina si un proyecto falla o no, pero, sin embargo, muestra el estado de los datos que están entrando a la base de datos, lo que permite dar un nuevo punto de vista a los interesados que estén usando el dashboard.
- Gracias al uso de MicroStrategy, se ha realizado un producto que permite la fácil adaptabilidad. Con las herramientas que ofrece, añadir nuevos módulos, modificarlos o borrarlos es una tarea sencilla. Además, como las consultas se basan en la creación y relación de objetos, añadir nuevos parámetros a gráficos ya realizados para obtener otros valores no conlleva a la necesidad de conocer todo el funcionamiento y proceso de creación, por lo que la aplicación podría ser mantenida por personas que desconozcan el desarrollo de la misma.
- La petición de obtener valores históricos para poder comparar los datos actuales con muestras anteriores para detectar si los patrones de comportamiento de los datos están tomando unos valores normales ha sido satisfecha. Mediante los filtros de MicroStrategy se pueden obtener los intervalos de datos pertinentes para mostrar en un mismo gráfico los subgrupos de información que se deseen.

Cabe destacar que las pruebas especificadas en el apartado 8 han servido para determinar ciertas tareas a realizar para conseguir un mejor resultado. Gracias a ellas, se ha conseguido detectar, por un lado, que la petición de carga del dashboard en ocasiones donde el trabajo a realizar por la base de datos es alto, tarda más de lo esperado (T. Espera > 10 segundos). Para solucionar esta problemática, se ha planteado la división del aplicativo en distintos dossiers, para que la carga de trabajo sea menor. De esta manera, solo se cargarán los datos que se están visualizando, evitando tener datos guardados de una pantalla en segundo plano que no se está utilizando. En las otras franjas horarias no existe tal problemática ya que las pruebas han dado valores por debajo del tiempo límite establecido en los requerimientos, pero gracias a la solución propuesta se conseguirá disminuir también los segundos de carga del dashboard.

La problemática principal encontrada en cuanto a la interfaz consiste en lo poco intuitivo de los datos en la pantalla principal. Esta es la más importante puesto que da una visión general del funcionamiento de todos los módulos y por lo tanto la información representada tiene que ser muy descriptiva. Para lograr solucionar este problema, se debe reestructurar la información mostrada, unificando algunos gráficos para que realizar comparativas no sea una tarea difícil. Concretamente, para los cierres de encargos se deben unificar los 3 posibles estados del cierre en una misma columna, para que de forma visual se puedan realizar comparativas. Para el caso de la apertura de encargos y pagos, se tienen que unificar los ejes para evitar escalas

distintas. lo que provoca que a simple vista la interpretación de los datos pueda ser incorrecta. En cuanto a la visualización de los umbrales, se ha decidido agregar los datos de la tabla que se estaba mostrando para tener un indicativo más genérico del estado, sin necesitar navegar en la grid para obtener información útil.

Además, gracias a los test de usabilidad se ha detectado otra problemática general, y esta es que no es lo suficientemente intuitivo el método de navegación, en el que se le tiene que hacer clic a un elemento del gráfico para navegar. Puesto que esta característica es algo que no se puede modificar ya que viene impuesto por la herramienta, se plantea la generación de documentación juntamente a la preparación de una sesión de formación para el equipo para entender el funcionamiento de la aplicación.

## 10 PLANIFICACIÓN

La planificación (Figura 8) ha sido modificada sustancialmente en 2 ocasiones. En primer lugar, es necesario comentar que por lo general la duración de las tareas de análisis y desarrollo se han alargado. El motivo principal es que, en un primer momento, la planificación no se pudo precisar de la manera debida, puesto que se realizó en una fase temprana de la vida del proyecto, donde todavía no se tenía conocimiento del tiempo necesario a invertir con exactitud. Como se puede observar, y a diferencia de planificaciones anteriores, también se ha aumentado el número de semanas disponibles para el proyecto. Esto se debe, principalmente, a los problemas generados por el estado de la situación en la que se desarrolla el trabajo a raíz del COVID-19 [11].

Esta situación ha provocado que toda la plantilla esté teletrabajando y que, por lo tanto, el nivel de comunicación haya disminuido. Esto no significa que no haya habido una proactividad a la hora de comunicarse, pero sin embargo ha alargado los tiempos de espera la cual cosa ha retrasado el avance del trabajo.

Además, decidir el uso de MicroStrategy como herramienta ha sido un proceso en el que han intervenido personas de distintos departamentos, y tal y como se ha comentado, obtener respuestas ha sido un proceso más largo de lo habitual. Además, desde el momento de la decisión hasta que se ha permitido el acceso a dicha herramienta, ha pasado una franja de tiempo de dos semanas (semanas 10 y 11 de la planificación) en la que el trabajo se ha parado y se ha trabajado exclusivamente en la documentación.

Por estos motivos, se ha decidido alargar unas semanas la vida del proyecto para poderlo finalizar de manera correcta. Por lo tanto, los tiempos para el análisis, desarrollo y pruebas se han retrasado unas semanas.

En cuanto a los recursos que han sido utilizados para el desarrollo de ciertas tareas a lo largo del ciclo de vida del proyecto, los más destacables son Visual Studio (para el análisis del código de BE y FE de las pantallas), Microsoft SQL Server Management Studio (para la creación de las consultas) y MicroStrategy (para la creación del FE y BE de la aplicación).

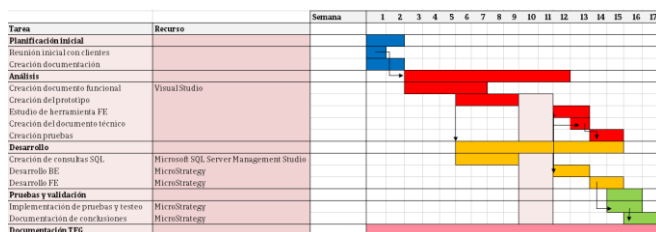


Fig. 8: Diagrama de Gantt del proyecto.

## 11 CONCLUSIONES

De manera general, cualquier tipo de empresa genera cantidades elevadas de datos cada día. Estos son guardados en registros de base de datos y consultados de manera esporádica, sin ser aprovechada toda la información que estos pueden llegar a dar.

Prepersa genera un alto volumen de datos nuevos diariamente. Consecuentemente, el departamento de desarrollo web de informática recibe a diario incidencias relacionadas con el uso de proyectos desarrollados por el equipo, que, de alguna manera u otra, están causando un problema al usuario.

Sin embargo, estas situaciones pueden ser previstas en función de como se están comportando los datos, del contenido de los registros que se generan diariamente en la base de datos de la empresa.

Teniendo en cuenta estas premisas, lo que este documento propone es dar solución a la entrada de incidencias gracias a la detección de situaciones anómalas que puedan concluir en errores para el usuario y la consecuente actuación.

Por lo tanto, el artículo explica la vida del proyecto que surge de esta problemática, realizando este trabajo con el objetivo de desarrollar un cuadro de mandos para la monitorización de datos generados y almacenados por Prepersa.

Se muestra el proceso seguido para la creación de la aplicación, destacando un análisis de los módulos que se deciden monitorizar y el uso de la herramienta MicroStrategy como framework para la implementación.

Gracias al aplicativo, la detección de problemas es más ágil, y se puede reaccionar proactivamente ante posibles situaciones de error, sin la necesidad de que este sea reportado por ningún usuario, dándole a los datos más valor puesto que están siendo usados para conocer el estado del funcionamiento del portal web. Sin embargo, la aplicación puede ser mejorada si se realizan las modificaciones planteadas en la sección de resultados, optimizándola y haciéndola más intuitiva.

## AGRADECIMIENTOS

Me gustaría agradecer este trabajo a tres personas. En primer lugar, a Jose Miguel Lorente y Andrei Fernando Hernández, compañeros del departamento de desarrollo informático de Prepersa e interesados en la creación de esta aplicación, por darme la oportunidad de desarrollar un proyecto desde su inicio hasta su fin.

Por otro lado, me gustaría agradecer el trabajo del profesor Victor Garcia, por su disponibilidad, consejos y apoyo durante la realización del trabajo.

## REFERENCIAS

- [1] "Seguros de coche, hogar, vida, salud, ahorro - Seguros Catalana Occidente", Segurosatalanaoccidente.com, 2020. [Online]. Disponible en: <https://www.segurosatalanaoccidente.com/>.
- [2] "Seguros de coche, hogar, vida, salud, ahorro - Plus Ultra Seguros", Plusultra.es, 2020. [Online]. Disponible en: <https://www.plusultra.es/>.
- [3] "Seguros de decesos, vida, hogar, ahorros - NorteHispana", Nor-tehispana.com, 2020. [Online]. Disponible en: <https://www.nortehispana.com/>.
- [4] "Seguros de coche, hogar, vida, salud, ahorro - Seguros Bilbao", Segurosbilbao.com, 2020. [Online]. Disponible en: <https://www.segurosbilbao.com/>.
- [5] Backend-to-frontend website development - Google Patents", Patents.google.com, 2020. [Online]. Disponible en: <https://patents.google.com/patent/US20160259509A1/en>, Mar 2020.
- [6] Godoy, D. (2015). Diseño de un Simulador Dinámico de Proyectos de Desarrollo de Software que utilizan metodología Scrum. "Gestión de Proyectos de Software: Metodologías Ágiles y Scrum". pp. 37-39, Feb 2020.
- [7] Godoy, D. (2015). Diseño de un Simulador Dinámico de Proyectos de Desarrollo de Software que utilizan metodología Scrum. "Gestión de Proyectos de Software: Metodologías Ágiles y Scrum". pp. 40-46, Feb 2020.
- [8] Dumas, S. Redish, C. (1999). A Practical Guide to Usability Testing. Disponible en: [https://books.google.es/books?hl=es&lr=&id=4lge5k\\_F9EwC&oi=fnd&pg=PR9&dq=Dumas+%26+Redish&ots=vrl78HbaBH&sig=VjROR-HUTzACHDDcsAbRItmsDW7s#v=onepage&q=Dumas%20%26%20Redish&f=false](https://books.google.es/books?hl=es&lr=&id=4lge5k_F9EwC&oi=fnd&pg=PR9&dq=Dumas+%26+Redish&ots=vrl78HbaBH&sig=VjROR-HUTzACHDDcsAbRItmsDW7s#v=onepage&q=Dumas%20%26%20Redish&f=false)
- [9] Sung Heum Lee. (1999). "Usability Testing for Developing Effective Interactive Multimedia Software: Concepts, Dimensions, and Procedures" [Online]. Disponible en: <https://www.jstor.org/stable/pdf/jeductechsoci.2.2.4.pdf>
- [10] Becker, Pablo y Olsina, Luis (2018) "Linking Business and Information Need Goals with Functional and Non-functional Requirements". [Online]. [Disponible en]: [https://www.researchgate.net/profile/Luis\\_Olsina/](https://www.researchgate.net/profile/Luis_Olsina/)
- [11] E. Dong, H. Du, L. Gardmer. (2020, mayo 01). "An interactive web-based dashboard to track COVID-19 in real time" [Online]. Disponible en: [https://www.thelancet.com/journals/laninf/article/PIIS1473-3099\(20\)30120-1/fulltext](https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(20)30120-1/fulltext)
- [12] "Home Prepersa", Prepersa.es, 2020. [Online]. Disponible en: <https://www.prepersa.es/inicio>.
- [13] Carpani, Fernando (2000) "CMDM: Un modelo conceptual para la especificación de bases multidimensionales". Pp 13, Abr 2020. Disponible en: <https://www.fing.edu.uy/inco/pedeciba/biblioteca/tesis/tesis-carpani.pdf>
- [14] Wixom, Barbara H. (2004) "Business Intelligence Software for the Classroom: MicroStrategy Resources on the Teradata University Network," Communications of the Association for Information Systems: Vol. 14, Artículo 12. pp 235 DOI: 10.17705/1CAIS.01412. [Disponible en]: <https://aisel.aisnet.org/cais/vol14/iss1/12>

## APÉNDICES

### A.1 Guión para los test de usabilidad

#### Introducción

¡Hola! Como ya sabrás, soy Guillermo y ahora mismo estamos reunidos para realizar unas pruebas con un producto que se está desarrollando. Durante la sesión, yo te iré guiando para realizar un test sobre un cuadro de mandos o dashboard que está sirviendo como proyecto de final de grado.

En primer lugar, me gustaría agradecerte que hayas ofrecido tu disponibilidad y disposición para ayudarme con este proyecto, todos tus comentarios y feedback aportado me ayudarán a saber si la aplicación funciona como se espera o no.

Ahora te comentaré los pasos que seguiremos durante la sesión. En primer lugar, te mostraré una pantalla en la que simplemente tendrás que explicarme que es lo que ves y que crees que es lo que se está mostrando. En segundo lugar, te pediré que realices unas tareas interactuando con la aplicación. Finalmente, te haré unas preguntas relacionadas con la experiencia que hayas tenido con el producto el cual se está probando.

Durante la sesión, te iré guiando, explicándote que es lo que tienes que hacer, pero no como hacerlo. Estaré también tomando notas de aquello que vaya pasando referente a tu interacción con el producto. Es muy importante que sepas que en ningún momento te estoy haciendo la prueba a ti, sino al producto. No vas a hacer nada mal. Si no sabes cómo hacer algo de lo que te pido, será un problema de la aplicación, y precisamente por eso estamos aquí, para que me puedas ayudar a detectar fallos para poder corregirlos. Por lo tanto, te pediría que pienses en voz alta, que todo lo que vayas haciendo lo expresas, sin miedo a realizar críticas. La clave de esta sesión es mejorar el producto, por lo tanto, me gustaría que seas lo más honesto/a posible con tus respuestas.

La prueba no tomará más de 20 minutos en realizarla y si en algún momento te apetece tomarte un descanso no dudes en decírmelo.

Dicho esto, ¿Tienes alguna duda? ¿Te gustaría comentarme algo antes de empezar con el testeo?

#### Observación

Bien, tal y como hemos comentado anteriormente, ahora te mostraré una pantalla. Por favor, no hagas ningún clic ahora. Me gustaría que la observaras y que me dijeras que es lo que ves y para que crees que sirve.

[Tiempo para la observación y comentarios].

#### Escenarios de prueba

Perfecto, estos comentarios me están dando un punto de vista totalmente nuevo sobre esta pantalla, te lo agradezco. De acuerdo, si te parece bien pasaremos a otro tipo de prueba. Ahora te plantearé unas situaciones en las que la respuesta puede ser encontrada utilizando la aplicación. Me gustaría que fueras navegando por ella hasta que encontrases la respuesta a los escenarios que te planteo. Y recuerda, si no encuentras la solución significará que el cuadro de mandos necesita unos ajustes.

#### • Escenario A (Cierre de encargos)

Nos gustaría saber cuánto ha tardado hoy a las 14:00, de media, en resolverse una petición de cierre para el módulo de autos.

#### • Escenario B (Apertura de encargos)

Imagina que nos comunican que muchos profesionales no están recibiendo encargos y por lo tanto no pueden trabajar. Quizá haya un problema relacionado con los encargos que se están abriendo. ¿Podrías buscar en el dashboard información al respecto?

#### • Escenario C (Pagos)

Bien, ahora me gustaría que me dijeras cuantos pagos han sido destinados a los profesionales de tipo Abogado y qué % representa sobre el resto de pagos.

#### • Escenario D (Mensajes)

El día 11/06/2020 sobre las 16:00 hubo un fallo en los servidores. Aparentemente nadie se ha quejado de ningún problema, pero algunos usuarios estaban esperando respuesta a un mensaje y todavía no la han recibido. Podríamos buscar si se perdieron mensajes durante el parón del servidor.

#### • Escenario E (Umbrales)

Existen casos en los que algunos peritos tienen mal asignados los encargos en base de datos, existiendo un desfase en su asignación. Me gustaría saber cuántos peritos del tipo particulares existen con este problema.

#### Preguntas finales

¡Perfecto! Ya hemos terminado con los casos, ahora simplemente te haré unas preguntas sobre la experiencia que has tenido, siéntete totalmente libre de contestar lo que quieras:

- ¿Si en la pantalla principal no pudieras hacer clic y solo pudieras observar los datos mostrados, crees que se podría obtener información útil de lo que se ve?
- ¿Crees que la pantalla ayuda a detectar situaciones en las que algo está fallando?
- ¿Te gustaría comentar algo que no te haya preguntado?

#### Cierre

Pues aquí finaliza el test. Permíteme una vez más agradecerte que hayas participado en la prueba y que hayas invertido parte de tu tiempo en este proyecto. Con tu participación he podido detectar ciertos aspectos a mejorar en el futuro de la aplicación. Tu aportación ha sido muy útil. ¡Gracias y hasta la próxima!