

# Software para la gestión y distribución de encuestas Opina Versión 2.8

Alejandro Herrera Barnes

**Resumen**— Opina es una aplicación web para la creación, gestión y distribución de encuestas. Cuenta con diferentes tipos de preguntas y es totalmente configurable gracias a la adición de mensajes de texto o a la organización de las preguntas dentro de grupos diferenciados. Con Opina se puede crear una encuesta, distribuirla entre contactos y ver las respuestas gracias a sus informes de resultados. Sus capacidades para exportar los resultados en diferentes formatos hacen a la aplicación ideal para su uso empresarial y obtener un feedback de calidad de los clientes o de los empleados. En este documento se exponen un conjunto de nuevos desarrollos que se han realizado para dotar a Opina de nuevas funcionalidades y así actualizar la aplicación a su nueva versión 2.8.

**Palabras clave**— Opina, Encuestas, IN2, Desarrollo de software, aplicaciones web, java, spring, Inversión de Control, metodologías ágiles, Scrum, programación orientada a objetos, ORM, javascript, ExtJs, frameworks

**Abstract**— Opina is a web application for the creation, management and distribution of surveys. It has different types of questions and is totally configurable thanks to the addition of text messages or to the organization of the questions within different groups. With Opina you can create a survey, distribute it between contacts and see the answers thanks to your results reports. Its capabilities to export results in different formats make the application ideal for business use and get quality feedback from customers or employees. This document presents a set of new developments that have been made to provide Opina with new functionalities and thus update the application to its new version 2.8.

**Index Terms**— Opina, Surveys, IN2, Software development, web apps, java, spring, Inversion of Control, agile, Scrum, Object-Oriented-Programming, ORM, javascript, ExtJs, frameworks

---

◆

## 1. INTRODUCCIÓN

Pocas veces se le da importancia a lo arraigados que están en nuestra sociedad unos elementos tan simples como las encuestas y los formularios. Si se reflexiona sobre este tema, se llega a la conclusión de que son unas herramientas extremadamente útiles a la hora de recolectar información, sea de un número reducido de escogidos o de una parte significativa de la población. ¿Qué sería de los gráficos de intención de voto previos a unas elecciones sin la multitud de encuestas que se realizan a la población?

En el ámbito empresarial, estos formularios también están muy extendidos, ya que pueden ser usados para objetivos tan dispares como conocer la satisfacción de un cliente respecto a la marca, conocer las inquietudes o hábitos de trabajo de una plantilla o tomar decisiones estratégicas en función de las respuestas y la información de negocio extraídas de las encuestas a un sector target de la población.

El modelo de encuesta tradicional, donde una o varias personas se pasan horas y horas en una concurrida calle asaltando a los viandantes con una hoja de papel, es un modelo que, por supuesto se sigue usando en la actualidad porque por pura estadística, la insistencia de los encuestadores hace que se consigan respuestas. Este modelo es útil para encuestas que tienen como temática alguna cuestión de interés general y en muchas ocasiones la contestación de estos cuestionarios viene precedida por la

entrega de un incentivo. Como es de esperar, con este modelo no se puede obtener la opinión de una plantilla de trabajo, por ejemplo, si ésta es grande o si la propia encuesta requiere que sea contestada por la totalidad de un colectivo. Aquí es donde entran en juego las tecnologías de la información y la comunicación como Internet. Mediante Internet, el área de recursos humanos de una gran compañía, por ejemplo, puede distribuir una encuesta sobre ambiente laboral al resto de áreas de la empresa y controlar quién la ha contestado y quién no, y tomar las medidas oportunas para forzar a que todos los miembros de la corporación den su opinión sobre el tema. Además, una ventaja del modelo de encuestas 2.0 es el de la extracción de datos, ya que el tipo de aplicaciones que lo hacen posible suelen incorporar funciones para conocer las respuestas de los encuestados y módulos estadísticos, desde muy básicos y simples a modelos más avanzados y complejos, que permiten extraer la información deseada y tomar las decisiones de negocio oportunas.

Opina es una aplicación web para la creación, gestión y distribución de encuestas propiedad de la empresa de tecnologías de información IN2, con sede en Barcelona. Con Opina se pueden crear encuestas y formularios de forma simple y distribuirlas entre los usuarios que se desee. Posee varios tipos de preguntas que permiten al usuario crear cuestionarios versátiles mediante una excelente experiencia de usuario. Se incluyen en este software

otras opciones realmente interesantes como obtener los resultados de las encuestas mediante una serie de informes, controlar que usuarios han respondido los cuestionarios y exportar las respuestas en varios formatos. El siguiente documento tiene como objetivo exponer las nuevas implementaciones y desarrollos que se han realizado en los últimos meses con el fin de dotar a Opina un salto evolutivo en cuanto a sus capacidades y conformar lo que es la nueva actualización del software, nombrada como Opina V2.8. Las funcionalidades y los resultados de su implementación, que se explican en este documento, son funcionalidades demandadas por clientes de IN2 y, por tanto, en un principio, esta actualización de software se desplegará en los entornos cerrados y privados de cada uno de los clientes que la requieran. En concreto, esta nueva versión está formada por 3 funcionalidades nuevas, que se comentarán en la sección Objetivos de este documento. Como excepción, existe la posibilidad de desplegar la funcionalidad 2 (Sistema de lógica condicional) en el entorno de producción de la empresa para que la versión abierta al público general pueda disfrutar de esta funcionalidad que dota a Opina de más inteligencia y de una mejora significativa en la comprensión de encuestas y en la experiencia de usuario.

El presente documento está dividido en las siguientes secciones:

- **Introducción:** Breve introducción a Opina y al trabajo realizado.
- **Estado del arte:** Situación actual del software y breve introducción a las tecnologías utilizadas.
- **Objetivos:** Objetivos principales y desglose de subobjetivos de este TFG.
- **Metodología:** Comentarios sobre la forma de trabajar para este proyecto en colaboración con el propietario IN2 y procesos realizados.
- **Tecnologías utilizadas:** Explicación de las tecnologías en las que está desarrollado Opina y el uso que el sistema hace de ellas.
- **Implementaciones:** Proceso y comentarios sobre las decisiones tomadas para satisfacer los objetivos y requerimientos de cada una de las funcionalidades.
- **Resultados:** Resultados obtenidos como consecuencia de la implementación de las diferentes funcionalidades.
- **Conclusiones:** Reflexiones sobre el desarrollo de aplicaciones web y sobre aspectos técnicos aprendidos durante la ejecución del trabajo.

## 2. ESTADO DEL ARTE

Actualmente existen múltiples sistemas de encuestas en el mercado, desde sistemas de uso completamente gratuito y enfocados al público general, hasta sistemas más cerrados propiedad de grupos u organizaciones como universidades o empresas. No existe un modelo único de sistema gestor de encuestas, ya que cada uno presenta alguna particularidad, pero todos se basan en el mismo principio,

crear encuestas, distribuirlas, hacer que los usuarios las contesten para sacar el máximo de información sobre el tema propuesto y analizar sus resultados. Prácticamente la totalidad de estos sistemas son servicios web, ya que esta plataforma permite que el encuestado no se tenga que instalar en su dispositivo ningún tipo de software cliente específico para contestar una encuesta que puede que se resuelva en menos de un minuto. Además, el punto de entrada a estas encuestas suele ser el mismo para este tipo de aplicaciones, una URL que bien se envía a los encuestados mediante algún mecanismo de comunicación externo al aplicativo o mediante una notificación vía correo electrónico si es que la aplicación lo soporta. Opina es un sistema más de este tipo, maduro y completo que actualmente está desplegado en un servidor abierto al público, pero también se encuentra instalado en los servidores privados de varias organizaciones clientes de IN2. Este hecho lo hace muy interesante, ya que el conjunto de usuarios que mantiene Opina es un grupo más cerrado y reducido y permite que se puedan desarrollar nuevas funcionalidades a medida en función del cliente que lo requiera y del contexto en el que se encuentre este cliente. Opina está desarrollado principalmente en JAVA, toda la parte de back-end, y apoyado por el framework de Inversión de Control (IoC) Spring. Spring es uno de los frameworks de este tipo más usados en la actualidad, ya que permite al desarrollador olvidarse del control de los beans y las dependencias y centrarse en el desarrollo de su aplicación. Junto con Spring, la otra gran tecnología que conforma el back-end de la aplicación es el software de Object-Relational mapping (ORM) Hibernate, que proporciona una capa de abstracción sobre las entidades de la base de datos y permite mapear estas entidades en objetos JAVA. En el lado cliente se encuentra la biblioteca de Javascript para la generación de contenido dinámico Ext JS de Sencha. Obviamente, además de estas tecnologías se usan otras que se comentaran en secciones posteriores de este documento.

Cabe destacar también el grado de madurez de la aplicación. Opina es un sistema completo y funcional que da servicio tanto al público general como a clientes particulares, como ya se ha expuesto anteriormente. Este hecho hace que detrás del producto exista un servicio de mantenimiento y solución de incidencias. Además, la aplicación se encuentra instalada en un servidor de desarrollo que se usa como banco de pruebas para las nuevas funcionalidades. Opina sigue un paradigma de programación Model-View-Controller (MVC) y un fuerte patrón de herencias entre las clases Java que forman el producto. Para extender el código e introducir nuevas capacidades es necesario un conocimiento profundo del producto, ya que dado a su tamaño considerable puede ser complicado entender ciertas acciones que realiza o encontrar en qué parte del código se ejecutan.

## 3. OBJETIVOS

El objetivo principal del proyecto ha sido el desarrollo completo, es decir, diseño, implementación, documentación, pruebas y despliegue de la nueva versión (v2.8) de

la herramienta web para la creación y la difusión de encuestas Opina.

Para la consecución de este objetivo ha sido imprescindible cumplir una serie de subobjetivos, que correspondieron a cada una de las funcionalidades a desarrollar y que forman la nueva actualización. Estos subobjetivos y sus correspondientes funcionalidades se describen a continuación:

1. Desarrollo completo de un sistema multilinguaje que sea capaz de detectar las preferencias de lenguaje del cliente y muestre los elementos que forman la encuesta en el idioma detectado de forma automática.
2. Desarrollo completo de un sistema de lógica condicional que permita al creador de la encuesta relacionar preguntas con respuestas a preguntas anteriores, dentro de la misma encuesta.
3. Desarrollo completo de un mecanismo que permita a un usuario dejar de compartir una encuesta compartida.

Es necesario contextualizar este objetivo antes de entrar en más detalle. Opina en versiones anteriores contaba con una capacidad para que los usuarios dieran acceso a la información de explotación de datos de sus encuestas. Esta funcionalidad quedó inacabada y faltaba desarrollar un flujo que permitiera al usuario que comparte sus encuestas, detener la compartición de las mismas y hacer que la información de dicha encuesta volviera a ser propiedad exclusiva del usuario propietario.

Para estos 3 subobjetivos, se ha establecido, marcado por la empresa, un orden de prioridades.

Dando la máxima prioridad al objetivo 1, que dota a la aplicación de un mayor grado de internacionalización y la hace más usable.

En segundo lugar, en cuanto a prioridad e importancia se refiere, se sitúa el cumplimiento del objetivo 2, que permite a la aplicación dar un paso hacia adelante ofreciendo a los clientes la posibilidad de crear encuestas más personalizadas y específicas.

Finalmente, y como menos prioritario, se encuentra el subobjetivo 3, que como resulta ser la finalización de un mecanismo incompleto que está funcionando, el tiempo de desarrollo y el coste de trabajo son inferiores que el del resto de objetivos a cumplir.

## 4 METODOLOGÍA

La metodología de trabajo que se ha seguido para este proyecto fue impuesta por la empresa IN2, y que es la que actualmente usa en todos sus proyectos, la metodología ágil conocida como ScrumBan (Scrum + KanBan).

Scrum es un marco de trabajo para la gestión y desarrollo de software basado en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

Para el desarrollo de cada uno de los subobjetivos comentados en la sección "Objetivos", se ha seguido el procedi-

miento que se describe a continuación:

1. Exploración del elemento a modificar: Instalación de puntos de interrupción y mensajes de log para trazar el intercambio de datos creados por el cliente desde su introducción hasta su inserción en base de datos. Mismo procedimiento para conocer las extracciones de la BD y envío de información al cliente.
2. Creación de clases de tipo modelo para la persistencia que cumplan los nuevos requerimientos.
3. Modificaciones en el client side de la aplicación.
4. Modificaciones en la recepción y el envío de datos en el server side, es decir, modificaciones en las API REST.
5. Procesado de información en el server:
6. Pruebas de tipo exploratorio de los elementos afectados por los cambios.
7. Muestra del elemento funcionando al Product Owner, que debe validar la corrección del elemento o subobjetivo.
8. Subida de cambios al repositorio del proyecto.
9. En caso de finalizar toda la funcionalidad y cumplir un objetivo de rango superior, prueba completa de la funcionalidad, generar ejecutables y despliegue en servidores de clientes.

Esta metodología ha venido impuesta por la empresa por lo que en los únicos puntos donde se ha ofrecido flexibilidad son el 7 y el 8, debido a que, si un cambio no es crítico, se pueden obviar ya que sería mostrado al Product Owner en la prueba de la funcionalidad general.

El hecho de implementar ya desde fases tempranas del proyecto el ciclo comentado anteriormente da al cliente (representado por el Product Owner) una idea de cómo se van comportando las funcionalidades esperadas sobre el sistema final, proporcionando un valioso Feedback que genera ideas imposibles de concebir en un primer momento.

## 5 TECNOLOGÍAS UTILIZADAS Y ARQUITECTURA

### 5.1 Arquitectura general

Opina es un proyecto que sigue el patrón MVC, como se ha mencionado anteriormente, y que está configurado de la siguiente forma.

Por cada acción que el usuario puede realizar con la aplicación existen:

- Clases de tipo *model*: Representación en forma de objeto Java de una entidad de la base de datos. Bean de Hibernate ORM del tipo @Entity.
- Clases de tipo *service*: Encargadas de usar los Objetos de Acceso a Datos (DAO), encargados del acceso a la base de datos, a su conveniencia. Bean de Spring tipo @Service.
- Clases de tipo *resource*: Objetos que sirven para representar los datos que cliente y servidor se envían durante las transacciones. Mapean su

contenido en objetos de tipo modelo que provoca que las operaciones de transformación de datos se realicen sobre estas clases tipo resource.

- Clases de tipo *processor*: Encargadas de realizar las operaciones de negocio sobre los objetos resource. Forman parte de la parte controller del modelo MVC. Son beans de Spring de tipo `@Component` y se encargan de usar los *service*, tratar los datos recibidos en los endpoints de la API y de instanciar las clases resource.
- Clases de tipo API: Forman la API rest de la aplicación y son los puntos de intercambio de datos entre cliente y servidor. Juntamente con los *processor*, formarían la parte de controladores de la aplicación, ya que su función es la de recibir y preparar los datos, tanto de cara al client-side, como de cara a los modelos.
- Vistas de la aplicación: Desarrolladas con el framework de Sencha ExtJs. Se encargan de generar dinámicamente los contenidos html sobre la JSP principal de la aplicación. Se entrará en más detalles en la sección "Lado cliente".

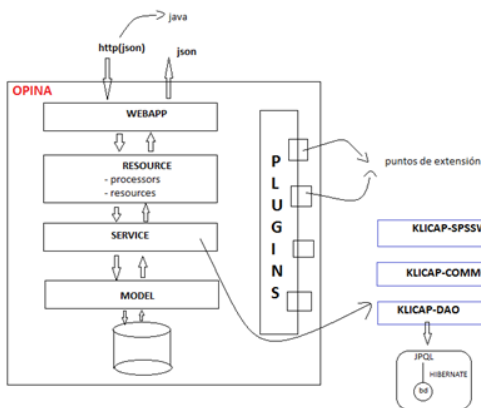


Figura 1: Arquitectura de Opina. Se observa la interacción entre los diferentes módulos que forman la aplicación para atender peticiones http del cliente.

## 5.2 Spring

Spring es un framework de IoC y un potente inyector de dependencias. Las dependencias y los objetos que se utilizan cuando una acción se inicia, normalmente por el cliente (de ahí lo de Inversión de Control), son inyectados por el contenedor de dependencias en tiempo de ejecución. De esta forma, el programador no tiene que ocuparse del ciclo de vida de los objetos, es decir ni los tiene que instanciar para crearlos, ni tiene la necesidad de guardar todos los objetos que usa cada cliente conectado a la aplicación web, técnica que en el desarrollo de aplicaciones de web es totalmente contraproducente e inútil. Spring, entonces, se encarga de construir los objetos que la aplicación necesita adaptándose al resto de frameworks que se usen en el lado servidor de la aplicación e integrando las diferentes clases de dichos frameworks, despreocupando al programador de la tediosa tarea de la gestión de los objetos. Al controlar Spring qué objetos debe inyectar, se evita que cada clase tenga que gestionar sus propias dependencias y por tanto conocer la implementación de las mismas, cosa que

produce un código fuertemente acoplado. Spring se encarga de automatizar y controlar tareas básicas en cualquier aplicación web como la gestión de sesiones, cookies y la seguridad de la aplicación. Además, provee de toda una interfaz para crear y gestionar APIs rest. El uso de Spring hace desaparecer de nuestro código todas las construcciones de objetos y se sustituye bien por un archivo XML de configuración de interdependencias, o bien por anotaciones en las clases que usamos, que es el modelo que sigue Opina.

```

@Service
@Path(ApplicationConfig.RESTFUL_PATH + "/survey")
public class SurveyAPI {
    @Autowired
    private SurveyProcessor surveyProcessor;

    @GET
    @Path("/{id}/simple")
    @Produces(MediaType.APPLICATION_JSON)
    @Security(value = { Role.ROLE_USER })
    public String getSurveyById(@PathParam("id") final Long id, @Context final HttpServletRequest request)
        throws MarshallException, ResourceException, NotFoundException, AccessDeniedException {
        String json;
        LOGGER.debug("Loading survey #" + id);
        SurveyResource surveyResource = surveyProcessor.load(id);
        try {
            json = JsonUtil.getBasicSerializer("survey").deepSerialize(surveyResource);
        } catch (RuntimeException e) {
            throw new MarshallException("survey.marshall.error", e);
        }
        return json;
    }
}

```

Figura 2: Ejemplo de clase SurveyAPI del package API, que, mediante anotaciones de Spring, crea un endpoint que responde a peticiones http get y devuelve la información de una encuesta dada su ID.

## 5.3 Mapping de objetos de bases de datos relacionales

Opina usa Hibernate para mapear el contenido de las entidades de la base de datos en Plain Old Java Objects (POJOs), formando así las clases modelo. Para acceder a los datos almacenados y hacer posibles las inserciones, eliminaciones y actualizaciones de la base de datos, se usan unos servicios conocidos como objetos DAO, que se encargan de manejar la base de datos y traducir su contenido a las clases modelo. Como se ha mencionado antes, Spring se encarga de gestionar el ciclo de vida de los beans de Hibernate. Hibernate fue montado sobre MySQL, que era el sistema gestor de base de datos que incorporaba Opina desde su primera versión.

## 5.4 Lado cliente

En el lado cliente de Opina hay que diferenciar dos vertientes:

### 5.4.1 Aplicación de creación/gestión de encuestas

Los formularios, los botones, las cuadrículas y el resto de componentes de la web son creados por una librería de Javascript llamada ExtJs propiedad de Sencha. En concreto Opina usa la versión 4.3 de esta librería que tiene como objetivo proporcionar una capa de abstracción y dando al programador la visión de que todos los componentes del front-end de su aplicación son objetos de javascript. Por tanto, como objetos, cuentan con propiedades personalizables que darán a cada componente unas cualidades únicas y unos comportamientos deseados. Estos contenidos se crean de forma dinámica y se montan sobre un JSP principal y los datos que muestran son recibidos del servidor mediante tecnología JSON. La librería es capaz de procesar estos objetos cuando se crean y traducirlos a

contenido HTML 5 normal, por tanto, su grado de personalización aumenta, aún más, si aplicamos CSS sobre ellos. Esta librería aplica también un patrón MVC que solo afecta al lado cliente y que se estructura con objetos *modelo*, encargados de mapear los contenidos de los formularios y los datos que se reciben del servidor. Los objetos *store* se encargan de la comunicación con la API; los objetos *view* son los componentes visuales de la aplicación y son los que albergan más parámetros de personalización y configuración; finalmente los objetos *controllers* hacen el binding entre los modelos y las vistas.

Esta librería es un tanto complicada de manejar por su infinidad de configuraciones, pero tener una buena comunidad de desarrolladores detrás y varios ejemplos resulta de gran ayuda.

#### 5.4.2 Página de contestación de encuesta

La página es contestación de la encuesta, visible sin necesidad de tener una cuenta en Opina, no usa la librería ExtJS, su funcionamiento es más simple. Se basa en una JSP donde los datos se montan mediante un preprocesador de HTML llamado Dust de Linkdn. Es el servidor el encargado de recuperar de la base de datos el contenido de la encuesta que se va a servir y usar el preprocesador para formar la página y enviársela, lista para renderizar al cliente. De esta manera, se desarrollan una serie de plantillas, que corresponden a los tipos de preguntas, a los bloques de texto y a los grupos de preguntas que aparecen en el cuerpo de la encuesta y es el servidor el que se encarga de hacer el binding entre datos y representación. Una vez esta página es renderizada por el cliente, es la librería de Google jQuery, la que se encarga de dar dinamicidad a los elementos y de crear los mensajes JSON con las correspondientes respuestas a cada una de las preguntas.

## 6 IMPLEMENTACIONES

### 6.1 Sistema de encuestas multilinguaje

Para resolver la situación que se planteaba en este objetivo, se desglosó la problemática en los siguientes grupos de tareas (no se incluyen subtareas por cuestiones de espacio):

1. Rediseño de la interfaz gráfica en la edición de preguntas
2. Mecanismos para detectar el idioma del navegador y enviarlo al servidor
3. Programación de funciones de las API, clases resource y clases processor
4. Creación de modelos en base de datos
5. Rediseño de los JSON que se intercambian cliente y servidor
6. Pruebas
7. Documentación: análisis técnico
8. Documentación: documentación durante el desarrollo
9. Documentación: actualización del manual de usuario

Cabe destacar, que, para implementar esta solución, y de igual forma que se hizo para el desarrollo del objetivo 2,

las implementaciones de la parte front-end y back-end se realizaron en paralelo, ya que era más sencillo generar la petición del cliente, y a partir de los parámetros enviados, empezar a programar toda la lógica del servidor. Primeramente, se observó que la base de datos no estaba preparada para soportar campos multiidioma. Un campo de tipo texto de una determinada entidad solo se podía guardar en un único idioma. En vez de alterar todas las tablas que tuvieran campos de texto que dependieran del usuario, se crearon nuevas entidades para representar estos elementos que pudieran guardar el mismo valor en 3 idiomas diferentes (catalán, castellano e inglés) y que se relacionaran con el registro de la tabla original mediante una clave foránea. De esta forma, cuando se fuera a consultar a la BD una pregunta, por ejemplo, se debían obtener los datos de la misma y luego consultar la tabla de las preguntas multiidioma para recuperar esos valores en el idioma correspondiente, en caso de tratarse de un acceso a contestación de encuesta, o en todos los idiomas disponibles, en caso de tratarse de un acceso para editar dicha pregunta. Este mismo procedimiento se aplicó a todos aquellos elementos de Opina que se pudieran visualizar en la página de contestación de encuesta, que són los enunciados, las validaciones, los mensajes de ayuda y las opciones de respuesta, en caso que tengan, de cada pregunta, grupos de preguntas, bloques de texto, título de la edición, mensajes de bienvenida y despedida de la encuesta y encabezados y pies de página. Tomando esta decisión, cada vez que un usuario define uno de estos elementos, se crea un registro en la tabla original y tantos registros como idiomas definidos en la tabla multiidioma correspondiente a ese elemento. A continuación, se expone un ejemplo:

El usuario crea dos preguntas, en una de ellas define los enunciados en los tres idiomas, en la otra, los define en dos idiomas. La tabla multiidioma de enunciados quedaría de la siguiente forma:

En la tabla original de preguntas, se crearían dos registros

ID pregunta (FK)	Wording	Idioma
25	¿Cuántos años tienes?	Es
25	How old are you?	En
25	Quants anys tens?	Ca
88	Valora la experiència de usuari	Es
88	Valora l'experiència d'usuari	Ca

Figura 3: Entidad OPINA\_MULTILINGUAL\_WORDING. Representa los enunciados de las preguntas en los diferentes idiomas.

con claves primarias iguales a las ID de la tabla anterior, el valor de uno de los enunciados, aquí se sigue un orden de prioridad siendo el primero el castellano, el segundo el catalán y el tercero el inglés, y el resto de datos de la pregunta. De esta forma, la tabla original queda inalterada y el mecanismo multiidioma es retrocompatible con elementos creados antes del despliegue de esta funcionalidad.

ID pregunta	Wording	Otros campos de la pregunta...
25	¿Cuántos años tienes?	
88	Valora la experiencia de usuario	

Figura 4: Entidad (versión reducida) OPINA\_QUESTION. Tabla que se encarga de guardar todos los datos de una pregunta. Se guarda el wording por defecto.

En la parte del front-end, se han modificado todos los formularios que tuvieran entrada de texto libre en la aplicación, eliminando el textarea simple e incorporando un selector que incluye tres textareas, uno por cada idioma, donde el usuario puede ir cambiando entre ellos sin que pierda que lo que ha escrito previamente. Esto implica que el servidor necesita más parámetros para llenar estas tablas, así que se han modificado los JSON sustituyendo los campos simples, por arrays de 3 posiciones donde viajaran los datos introducidos por usuarios. Cada posición del array contiene el idioma en que está escrito ese campo. Una vez estos mensajes llegan a la API, el servidor usa los nuevos recursos y los nuevos processors creados para acabar guardando esta información en las entidades persistentes. Será responsabilidad del creador de la encuesta definir todos los datos de los diferentes elementos en los mismos idiomas.

La otra parte de este desarrollo es la contestación de la encuesta. En el momento que un usuario accede, a través de la URL de encuesta, al cuestionario, mediante jQuery se detecta el idioma del navegador y se envía al servidor. El servidor es el encargado de buscar la encuesta demandada, y consultar todas las tablas multiidioma en busca de los elementos que la componen en el idioma del navegador recibido. En caso que se acceda con un idioma no soportado en Opina, se mostrarán los elementos de la encuesta en el idioma por defecto, siguiendo siempre las prioridades castellano, catalán, inglés. Si solo se han definido los elementos en un idioma, bien por decisión del autor, o bien porque son encuestas que se han creado antes del despliegue del sistema multiidioma, se mostrarán los textos en el idioma que haya en la tabla original.

## 6.2 Sistema de lógica condicional

Para resolver el objetivo 2, se ha usado el siguiente desglose de tareas:

1. Programación front end. Grupos condicionales
2. Programación front end. Formulario para relacionar opciones y grupos condicionales + comunicación JSON con el servidor
3. Programación lógica de servidor. Comunicación JSON con el cliente + reprogramación de Resources y processors
4. Creación de modelos en base de datos
5. Programación Front end. Visualización de la pantalla de encuestas
6. Pruebas
7. Documentación: análisis técnico

8. Documentación: documentación durante el desarrollo
9. Documentación: actualización de manual de usuario

En esta solución, igual que en la anterior, también se distinguen dos casuísticas diferentes. Cómo gestionar las relaciones entre preguntas cuando el usuario crea la encuesta y como mostrar las relaciones entre las preguntas en el momento de contestarla. Antes de empezar la explicación de la toma de decisiones hay que destacar que únicamente las preguntas de tipo “Múltiples Opciones” son las que pueden contener relaciones con otros elementos de la encuesta. El requerimiento seguido es que una pregunta que tiene, por ejemplo, 3 opciones diferentes, puede estar relacionada como máximo a otros 3 componentes del diseño de la encuesta, que pueden ser preguntas, bloques de texto o grupos de preguntas. Al ser los bloques de texto y las preguntas unidades independientes, se decidió crear un nuevo tipo de elemento llamado grupo condicional. Este nuevo tipo de grupo puede contener preguntas, textos y otros grupos, pero a diferencia de los grupos de preguntas tradicionales, no tiene aspecto en el cuerpo de la encuesta y por tanto no se muestran hasta que la opción a la que van asociada es marcada. Los grupos condicionales, a ojos del servidor, son una unidad, por tanto, todo lo que se incluya dentro de un mismo grupo condicional podrá ir asociado a una opción de respuesta. De esta forma, un creador podrá indicar que la “opción A” de su pregunta 1, estará relacionada con las preguntas 2, 3, 4 y el mensaje de texto t1 y que éstos solo aparecerán cuando el encuestado marque dicha opción. Los grupos condicionales son objetos JAVA que heredan del grupo base, lo único que los diferencia es su aspecto a la hora de ser renderizados. Para recrear esta situación se ha añadido un flag, llamado elementType, en la entidad SurveyElementSet para distinguir qué grupos son normales, y tienen aspecto y descripción, y qué grupos son condicionales, que tienen otro estilo en las páginas de creación de encuesta, carecen de descripción y no se renderizan en la página de contestación del cuestionario.

ID Clave Primaria	Título (nombre del grupo condicional)	Descripción	elementType
2000	Preguntas de la opción “Sí”	null	conditional
2001	Preguntas de la opción “No”	null	conditional
2002	Preguntas sobre satisfacción en el puesto de trabajo	Contesta las siguientes preguntas sobre satisfacción en el puesto de trabajo	standard

Figura 5: Entidad OPINA\_SURVEY\_ELEMENTSET. Guarda los grupos de preguntas. Se observa el nuevo campo elementType.

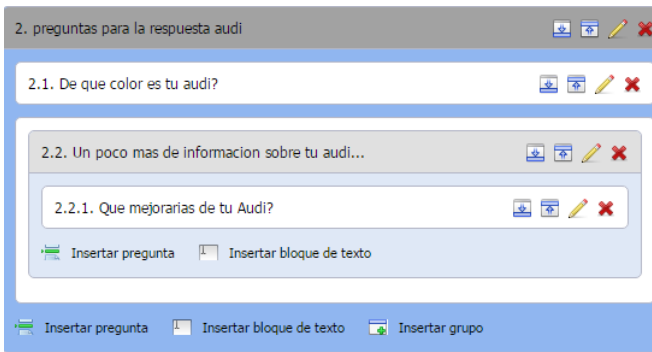


Figura 6: Grupo condicional (el azul oscuro) en la página de "Diseño" de la encuesta. Contiene una pregunta y un grupo normal que a su vez tiene otra pregunta.

Una vez teníamos estos grupos, surge la necesidad de almacenar las relaciones. Para ello era imprescindible guardar en el mecanismo de persistencia los siguientes valores:

- Pregunta de tipo "Múltiples Opciones" que inicia la relación.
- Opción de la pregunta que se implica en la relación.
- Grupo condicional.

La entidad recibió el nombre de OPINA\_LOGICAL\_RELATIONSHIP y tiene el siguiente aspecto:

ID (Clave Primaria)	ID Opción respuesta (FK)	ID Grupo condicional (FK)	ID Pregunta (FK)
780	501	2000	8555
781	502	2001	8555
782	522	2002	8555

Figura 7: Nueva entidad, OPINA\_LOGICALRELATONSHIP. Relaciona las opciones de respuesta con los grupos condicionales.

Paralelamente al desarrollo de la parte del servidor, se hicieron modificaciones en los formularios que permitían definir las opciones de las preguntas de múltiples opciones. Este formulario contenía una tabla donde el usuario introducía cada una de las posibles respuestas que podía tener la pregunta. Se añadió un nuevo campo a esta tabla que mediante una llamada AJAX al servidor, muestra con un selector desplegable los grupos condicionales creados para esa encuesta. El usuario crea la opción y puede escoger uno de los grupos condicionales creados, si es que quiere usar el sistema de lógica condicional. Este mecanismo implica que la manera de crear los cuestionarios en Opina quede más limitada, pues para usar este sistema correctamente, el usuario primero debe crear todas las preguntas, luego pensar y organizarlas en los grupos condicionales que le interesen y finalmente ir a las preguntas "padre", entrar en su configuración y crear sus opciones y relaciones.

La otra parte del desarrollo es la gestión de las relaciones en la página de encuesta. Todos los grupos condicionales creados, no deben mostrarse en el momento de acceso a la encuesta. En el momento en que el encuestado selecciona una opción que tiene una relación, se muestra el contenido del grupo condicional. Si esta opción se deselecciona o se pulsa el botón que tiene cada pregunta "Borrar datos", que limpia las opciones marcadas, el grupo relacionado debe ocultarse de nuevo. Para ello se ha tenido que modificar el JSON que recibe el encuestado añadiendo las relaciones entre las opciones de respuesta y los grupos que desencadenan. Estos nuevos grupos, además, incorporan un campo en el JSON que los identifica como grupos condicionales y que se usa para decidir la clase de CSS que implementan, cuyo comportamiento es no mostrarse por defecto.

Para crear los efectos de aparición y desaparición, se han modificado las plantillas que forman las encuestas. Como se ha comentado anteriormente, para formar el cuerpo de la encuesta se usa un preprocesador de HTML llamado Dust. Estas plantillas permiten incorporar tags javascript y mediante el uso de estos tags y la librería jQuery, se ha conseguido que la encuesta "escuche" los cambios que se producen en ella y en función de las opciones que se van marcando/desmarcando mostrar u ocultar el contenido relacionado. En caso que alguna pregunta de las que están ocultas tenga validaciones, como por ejemplo que sea obligada su contestación, se ha decidido suprimir dichas validaciones, ya que realmente una pregunta que no se muestra no tiene que ser contestada y por tanto la validación carece de sentido hasta que se enseñe la pregunta.

### 6.3 Finalizando la funcionalidad: Compartir encuesta

Para la solución del objetivo 3, no fue necesario planificar las tareas de forma tan minuciosa como en los desarrollos anteriores. Se identificó que la funcionalidad que permitía compartir encuestas estaba inacaba y que se podía compartir una encuesta, pero no detener la compartición. Para la implementación no ha sido necesario modificar la entidad OPINA\_SHAREDSURVEY, que era la encargada de almacenar las relaciones entre las encuestas y los usuarios que tenían acceso de explotación a ellas. Se ha usado el mecanismo normal de borrado de registros en Opina, que no es otro que marcar en la tabla donde se encuentran todos los identificadores de todos los objetos de Opina, la entrada correspondiente a la relación como eliminada. Para hacer esto, una vez que el usuario da de baja una pregunta, una respuesta de un cliente o un contacto, Opina añade la fecha de eliminación del elemento a esta tabla, conocida como OPINA\_DESKTOPELEMENT. Las relaciones entre encuesta y usuarios con acceso de explotación, heredan de este objeto por lo que también cuentan con una fecha de borrado. Así pues, lo único que se debió gestionar fue el proceso para indicar a Opina que borrase/invalidase un determinado registro. Para conseguir esto, se implementaron clases resources y processors y se creó un nuevo endpoint en la API que es a donde el cliente debía hacer las peticiones de borrado.

En la parte frontal, se añadió un botón al formulario de compartición de encuestas que permite, previa selección de los usuarios con los que no se quiere compartir, denegar el acceso a los mismos. Cuando se deniega el acceso a una encuesta que antes era compartida, ésta ya no se muestra en la pestaña “Encuestas compartidas” del otro usuario y por tanto no puede acceder a los informes del cuestionario.

## 7 RESULTADOS

### 7.1 Resultados del sistema de encuestas multidioma

Se obtiene un rediseño de las interfaces de usuario en las ventanas emergentes de creación de preguntas, bloques de texto, grupos de preguntas y en la pestaña de ediciones que permite introducir todos los campos de texto en los idiomas castellano, catalán e inglés. La base de datos es capaz de almacenar los campos de todos estos elementos en estos tres idiomas, pero manteniendo la retrocompatibilidad con el modelo anterior que solo contemplaba un único idioma por elemento.

El sistema es capaz de detectar el idioma del navegador del encuestado y mostrar los elementos de la encuesta en el idioma configurado, si es que está disponible. Si el idioma del cliente no está soportado por Opina o el creador del cuestionario no ha definido los elementos para ese idioma, se aplica un orden de prioridades y se entrega al cliente una encuesta con el idioma disponible.

### 7.2 Resultados del sistema de lógica condicional

Se obtiene un rediseño en la pestaña “opciones de respuesta” de las preguntas de tipo “Múltiples Opciones”, donde ahora se puede definir con qué conjunto de preguntas se asocia cada una de las opciones.

Se introduce el nuevo tipo de grupo de preguntas cuyo contenido solo será visible cuando se marque la opción, en la encuesta, que tiene asociada.

La página de encuesta es capaz de mostrar y ocultar contenido (otras preguntas, bloques de texto y otros grupos) en función de las respuestas anteriores de un usuario.

Gracias a esto, el usuario que va a explotar los resultados de la encuesta puede hacer foco sobre unos usuarios en concreto que han respondido de una forma una determinada pregunta, y el usuario encuestado evita visualizar preguntas que no tienen relación con él y por tanto hace la encuesta más usable y clara.

### 7.3 Resultados de la finalización de la funcionalidad “compartir encuesta”

Se obtiene la capacidad de dejar decidir al usuario con que usuarios se comparte o no una determinada encuesta, todo desde el mismo menú emergente de “Compartir encuesta”.

Los resultados de las funcionalidades 1 y 2, serán mostrados en los apendices de este documento en formato de captura de pantalla.

El código de cada una de las funcionalidades desarrolladas durante este proyecto, ha sido etiquetado y subido a tres ramas diferentes del repositorio privado de la empre-

sa IN2. Queda fuera del alcance del proyecto, la integración de estas funcionalidades en una única versión del programa.

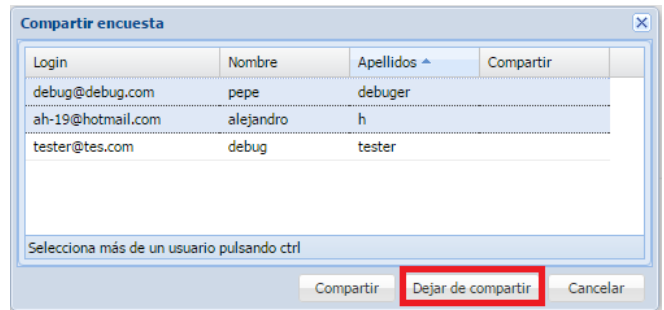


Figura 8: Ventana para compartir, y descompartir, una encuesta. En rojo el botón que inicia la funcionalidad descrita. Resultados visuales del desarrollo.

## 8 CONCLUSIONES

A raíz de la realización del proyecto y de la consecución de cada uno de los objetivos propuestos se extrae el siguiente conjunto de conclusiones, que se separan en apartados según su naturaleza:

### 8.1 Sobre la metodología y planificación

La metodología Scrum ha permitido reaccionar rápidamente a cambios en los requisitos, que han ido surgiendo a lo largo del desarrollo de los objetivos mediante la comunicación completa y directa con el *Product Owner*, que al tratarse de una empresa de pequeño-medio tamaño, también aporta opciones técnicas sobre el desarrollo del producto.

La metodología Scrum evita mal entendidos en los requisitos mediante las reuniones y revisiones constantes del trabajo.

Una planificación realizada con la suficiente antelación y con nociones técnicas del trabajo a realizar resultan muy orientativas y no sufren un gran número de modificaciones. En este aspecto, cabe destacar que la consecución del objetivo 1 fue, en cuanto a número de horas se refiere, algo corta y que la consecución del objetivo 2 se ha realizado en menos tiempo de lo planificado, lo que ha permitido invertir ese tiempo de más en el anterior objetivo.

La realización de análisis funcionales, encarados a la venta del producto y a exponer las funcionalidades con un lenguaje nivel usuario, y de análisis técnicos previos, da una doble perspectiva del trabajo a realizar, lo que permite tener más claro cuáles son cada uno de los subobjetivos y cuánto tiempo/esfuerzo se debe invertir en cada uno de ellos.

### 8.2 Sobre los aspectos técnicos

La arquitectura cliente-servidor produce un cambio en la forma de pensar el código y los diferentes algoritmos que se tienen que programar, respecto a la forma de hacer software de escritorio y cerrado a conexiones. El programador debe implementar el front-end de forma que le sea más fácil integrarlo con el back-end y a la vez debe tener

en cuenta la experiencia de usuario y el rendimiento que puede ofrecer el servidor donde se ejecute el servicio, ya que tendrá que tener en cuenta que ciertas operaciones, como, por ejemplo, introducción masiva de datos en el servidor, no son viables si se tiene a un usuario con un navegador web esperando al otro lado de la conexión. Esta arquitectura es mucho más orientada a eventos y a la interacción con el usuario que un programa que se dedique a procesar datos o a realizar cálculos matemáticos. Es por eso que el implementador se debe acostumbrar al trabajo con frameworks de inversión de control como Spring.

Spring dota a las aplicaciones web de una capa de seguridad y de un control de las peticiones al web service que permite al programador centrarse en el desarrollo de su aplicación y no en temas como la gestión de sockets o la gestión manual de las sesiones de usuario y cookies. Además, el framework incorpora compatibilidad con multitud de herramientas como parsers de JSON y compatibilidad con loggers como Log4j, básicas en el desarrollo de webapps. Es un potente sistema de inversión de control y un sistema de inyección de dependencias pensado para grandes aplicaciones web, ya que su funcionamiento y capacidades son excesivas para pequeñas aplicaciones. La experiencia con Spring, por lo general no ha sido mala, ya que la parte del inyector de dependencias obliga a seguir un patrón de programación que se repite a lo largo de toda la aplicación y que, ejecutado convenientemente, hace que el código funcione con el comportamiento deseado.

El ORM Hibernate facilita la gestión de la base de datos evitando el uso, en la medida de lo posible, de SQL. El mapeo de tablas en objetos es extremadamente útil, ya que permite alterar o crear registros solo con modificar el objeto JAVA que los representa en la aplicación. La definición de constraints o de relaciones entre tablas se facilita si se sigue correctamente las instrucciones de la documentación de Hibernate.

La herencia JAVA en proyectos con gran cantidad de clases, muchas veces hace difícil su comprensión y mantenimiento, por lo que se debe trabajar con mucho cuidado en este aspecto si no se quiere “estropear” nada de lo que ya está implementado. En este caso concreto, ninguna parte del código estaba pensada para ser extendida y obtener un sistema multilenguaje como el que se ha desarrollado en el objetivo 1, por lo que ha resultado complicado seguir la herencia en algunos puntos y se ha optado por generar clases, a veces muy similares entre ellas, con funciones muy específicas. La gestión de los mensajes JSON era otro de los mecanismos de difícil extensión, por lo que en varias ocasiones se han creado nuevos parsers de este lenguaje para tratar la información.

En cuanto a los frameworks de front-end, por una parte, se ha trabajado con Ext js de Sencha, que crea los compo-

nentes de la página web de forma dinámica y obtiene los diferentes datos a cargar mediante la comunicación JSON con el servidor. Cabe destacar, que quizá no es el framework para vistas más avanzado del mercado, cumple su función, pero no es demasiado intuitivo por lo que a veces resulta costoso y se pierde tiempo creando componentes tan simples como un desplegable, que en otro framework como Angular 2 o Liferay serían más fáciles de construir. Por otra parte, combinando el preprocesador de HTML, Dust de LinkedIn con jQuery para renderizar y controlar la página de encuesta, se obtiene una experiencia de usuario muy óptima y al programador le es muy fácil alterar los comportamientos y conocer lo que está pasando en todo momento.

## AGRADECIMIENTOS

Doy las gracias a mi empresa, IN2, por ofrecerme este trabajo y en especial a Naiara Suances (jefa de proyectos), Alex Ugartondo (jefe de producto) y Ana Cachaldora (analista) por ayudarme y ponerme facilidades durante el desarrollo.

Agradecer también a Juan Carlos Sebastián (tutor del TFG) por sus orientaciones y por dejarme trabajar a mi aire.

Finalmente, pero no menos importante, a mi familia y a mi pareja, Nerea, por aguantar mis charlas técnicas, que sé que tan aburridas les resultan.

## BIBLIOGRAFÍA

- [1] "Opina: Online Survey Manager", Opinahq.com, 2017. [Online]. Available: <http://opinahq.com>. [Accessed: 12- Jun- 2017]
- [2] "IN2 | Blog", In2.es, 2016. [Online]. Available: <http://www.in2.es/es/blog/opina-nueve-tipos-de-preguntas>. [Accessed: 12- Apr- 2017]
- [3] P. Gupta and M. Govil, "Spring Web MVC Framework for rapid open source J2EE application development: a case study", International Journal of Engineering Science and Technology, vol. 2, 2010.
- [4] L. Vlad Mihalcea, "Hibernate ORM 5.2.10.Final User Guide", Docs.jboss.org, 2017. [Online]. Available: [https://docs.jboss.org/hibernate/orm/current/userguide/html\\_single/Hibernate\\_User\\_Guide.html](https://docs.jboss.org/hibernate/orm/current/userguide/html_single/Hibernate_User_Guide.html). [Accessed: 15- May- 2017].
- [5] R. Johnson, J. Hoeller, K. Sampaleanu, R. Harrop and Others, Spring Framework Reference Documentation. Pivotal, 2016 [Online]. Available: <https://docs.spring.io/spring/docs/current/spring-framework-reference/pdf/spring-framework-reference.pdf>. [Accessed: 03- Jun- 2017]
- [6] "Ext JS 4.2.1 - Sencha Docs", Docs.sencha.com, 2017. [Online]. Available: <http://docs.sencha.com/extjs/4.2.1/>. [Accessed: 09- Jun- 2017].
- [7] "Dust.js | Getting Started", Dustjs.com, 2017. [Online]. Available: <http://www.dustjs.com/guides/getting-started/>. [Accessed: 12- May- 2017].
- [8] "jQuery API Documentation", jQuery API Documentation, 2017. [Online]. Available: <https://api.jquery.com/>. [Accessed: 07- May- 2017].

## APÉNDICES

A continuación, se muestran capturas de pantalla de la aplicación que muestran cómo queda el front-end de la misma después de haber desarrollado las funcionalidades 1 y 2.

### A1. RESULTADOS DEL SISTEMA DE ENCUESTAS MULTILENGUAJE

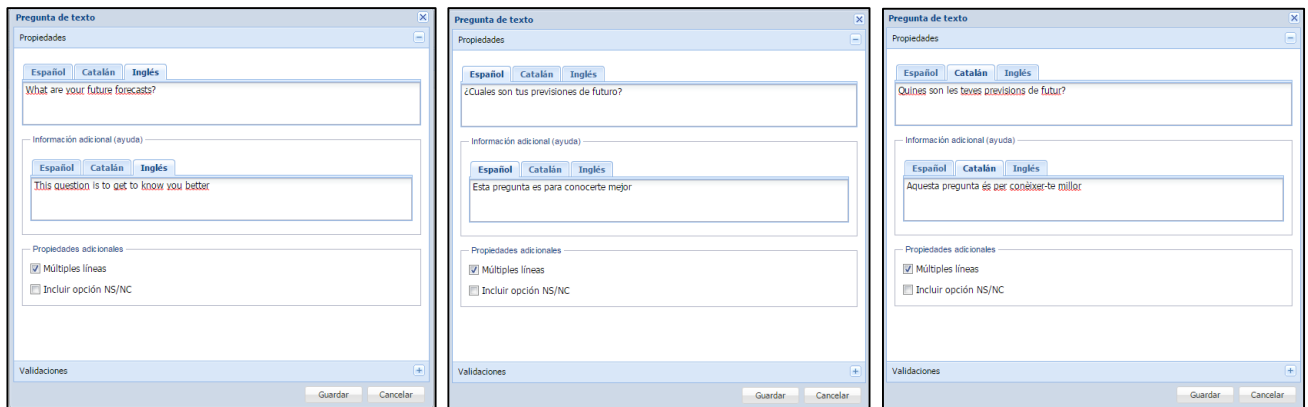


Figura 1: Ventana de creación/edición de pregunta. A la izquierda definiendo el idioma del enunciado y del mensaje de ayuda en inglés, en el centro en castellano y en la derecha en catalán.

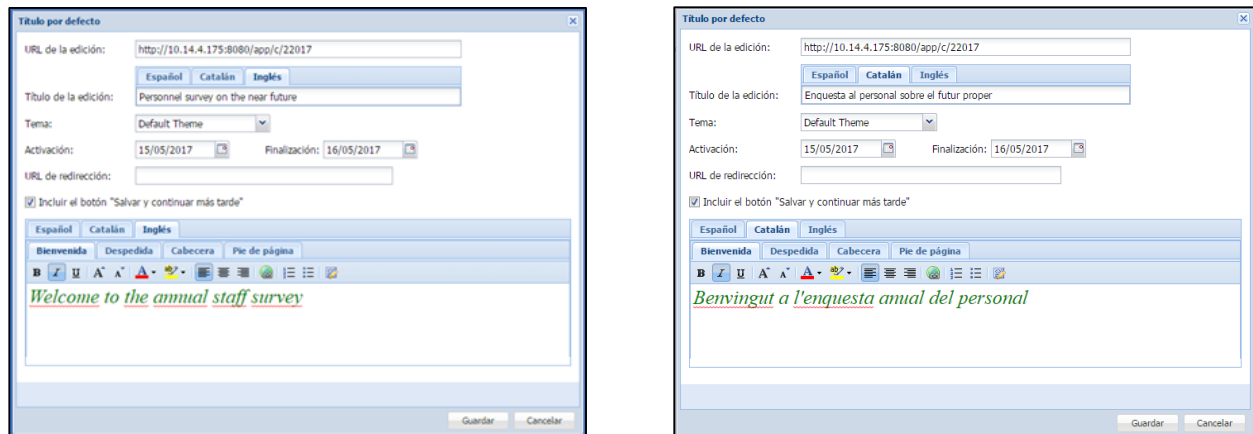


Figura 2: Interfaz de definición de los elementos de la publicación. A la izquierda, definiendo el título y el mensaje de bienvenida en inglés, a la derecha en catalán.



Figura 3: Misma encuesta acciando con el navegador en inglés, arriba a la izquierda, y accediendo con el navegador en catalán, arriba a la derecha.

## A2. RESULTADOS DEL SISTEMA DE LÓGICA CONDICIONAL

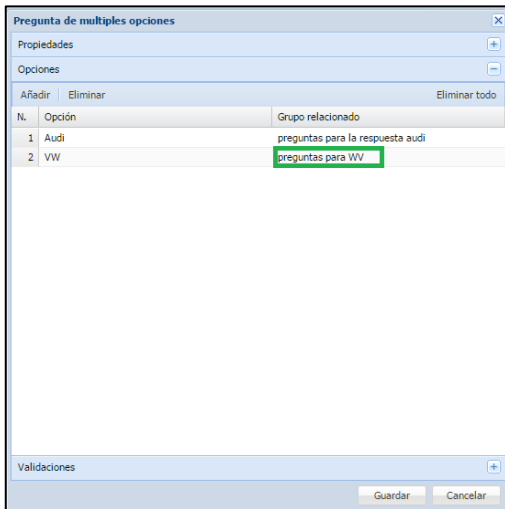


Figura 4: Ventana de asignación de opciones de respuestas con grupos condicionales.

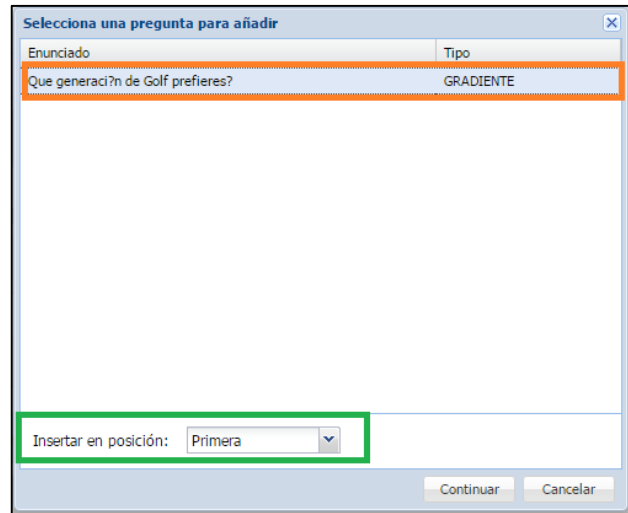


Figura 5: Ventana de adición de preguntas a grupo condicional

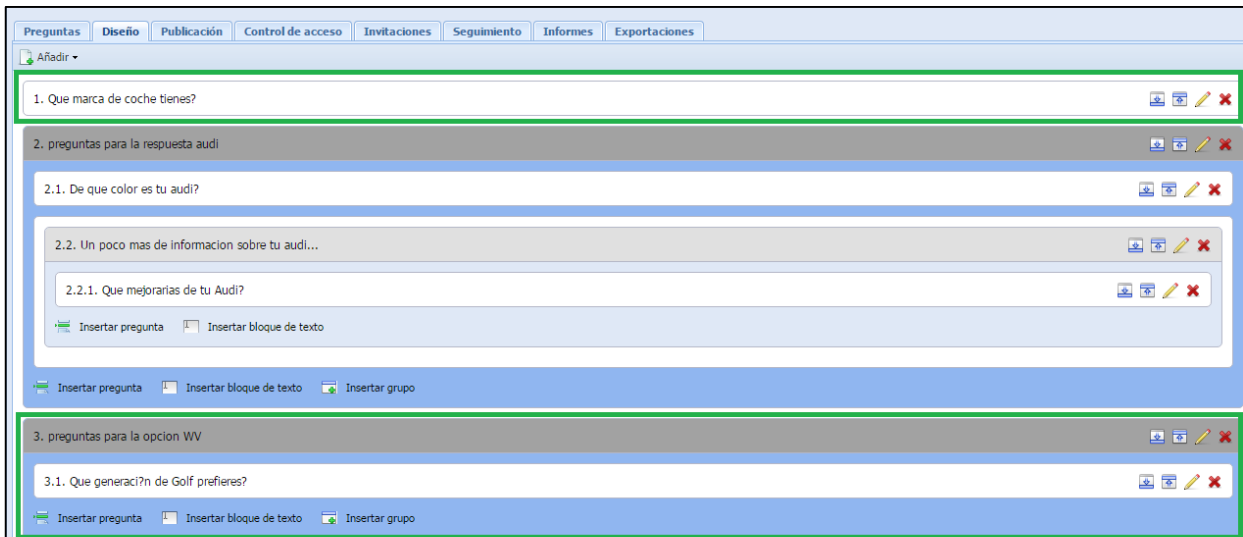


Figura 6: Pestaña de "Diseño" de la encuesta. Se pueden observar una pregunta de múltiples respuestas que tiene relación con los dos grupos condicionales de abajo. Cada grupo condicional contiene sus propios elementos.

**logica grupo WAG**

Borrar datos

Que marca de coche tienes?

Audi  
 Seat  
 VW

Borrar datos

De que color es tu audi? \*

**Un poco mas de informacion sobre tu audi...**

Contesta la siguiente pregunta

Borrar datos

Que mejorarias de tu Audi? \*

NS/NC

finalizar

salvar y continuar más adelante

Figura 7: Contenido de la encuesta al marcar la opción "Audi" en la primera pregunta.

**logica grupo WAG**

Borrar datos

Que marca de coche tienes?

Audi  
 Seat  
 VW

**Sobre la satisfacción con tu coche**

Borrar datos

Estas satisfecho con tu Seat? \*

Sí  
 No

**Otros datos importantes del vehiculo**

Borrar datos


Cuantos años tiene tu Seat? \*

finalizar

salvar y continuar más adelante

Figura 8: Contenido de la encuesta al marcar la opción "Seat" en la primera pregunta.


## logica grupo WAG

 Borrar datos

Que marca de coche tienes?

- Audi
- Seat
- VW

Seguro que tienes un Golf

 Borrar datos

Que generaci?n de Golf prefieres? \*

1 2 3 4 5 6 7

Powered by Opina: Online Survey Software • A product of IN2

Figura 9: Contenido de la encuesta al marcar la opci?n "VW" en la primera pregunta.