

SIRECA

Estudio e implementación de un Simulador de Replanteo de Catenarias

Adrián Aguirre Sánchez

Resumen— Hoy en día se realizan muchos proyectos alrededor del mundo, relacionados con el diseño y la construcción de catenarias para trenes, estos proyectos son dirigidos por empresas especializadas en el sector, como es Sener Ingeniería y Sistemas S.A. Este proyecto realizar un estudio y una implementación para desarrollar una aplicación web híbrida que trabaje con un núcleo el cual realiza tareas relacionadas con replanteos de catenarias, este software será capaz de automatizar el proceso que actualmente es manual y poder centralizar este servicio dentro de la empresa. Las simulaciones y pruebas de replanteos relacionadas con las catenarias son muy costosas, por ese motivo nace SIRECA, un proyecto innovador y con visión de futuro, el cual pretende mejorar los proyectos de catenarias venideros, de forma que logre reducir el tiempo y por lo tanto reducir los recursos y costes por parte de la empresa.

Palabras clave— Catenaria, ASP.NET, Bootstrap, HTML5, MVC, Subversion, MySQL, Visual Basic for Applications, Replanteo, Dibujado.

Abstract— Today many projects around the world, related to the design and construction of overhead lines for trains are made, these projects are managed by specialized companies in the sector, such as Sener Engineering and Systems S.A. This project is a study and an implementation to develop a hybrid web application that works with a core which performs related stakeout catenary tasks, this software will be able to automate the process that is currently manual, and to centralize this service within the company. Simulations and tests stakeout related to overhead lines are very expensive, for this reason born SIRECA, an innovative and visionary project, which aims to improve projects coming catenary, so that manages to reduce the time and therefore reducing resources and costs by the company.

Index Terms— Catenary, ASP.NET, Bootstrap, HTML5, MVC, Subversion, MySQL, Visual Basic for Applications, Stakeout, Drawn.



1 INTRODUCCIÓN

Sener [1] es una gran empresa que se ha dedicado a la Ingeniería en prácticamente todo su abanico de posibilidades, siempre enfocándose en la investigación, el desarrollo y sobretodo la innovación, de eso trata en esencia SIRECA.

SIRECA es un proyecto de desarrollo interno de la empresa Sener, pretende ser esa herramienta que pueda ser

utilizada por cualquier ingeniero eléctrico para proyectos de forma interna para facilitar las tareas relacionadas con los proyectos de catenarias [2] de ferrocarriles. SIRECA, entre otras características, gestionará, mantendrá y mostrará todos los datos relacionados con la catenaria seleccionada, estos datos podrán ser manipulables para realizar nuevas configuraciones, también será capaz de realizar cálculos de replanteo utilizando los datos almacenados junto a información cedida por el usuario, otra utilidad o uso de SIRECA es la función de pre-dibujado, que es un tramo de toda la catenaria, o dibujado completo de un plano bidimensional de la catenaria.

- E-mail de contacte: adrian.agui9@gmail.com
- Menció n realizada: *Tecnologías de la Información.*
- Trabajo tutorizado por: *Ramón Martí Escalé (DEIC)*
- Curso 2015/16

La aplicación pretende tener una interfaz intuitiva para su uso, enfocándose en ingenieros eléctricos que vayan a usarla y no en usuarios genéricos, esto quiere decir, que la aplicación requerirá de un conocimiento previo sobre el sistema eléctrico de una catenaria para proceder a su uso.

Cabe destacar que SIRECA es una aplicación web la cual pretende funcionar de manera centralizada para dar servicio a toda la empresa desde un servidor virtual ya preparado para alojar la aplicación web.

SIRECA no solo se centra en una aplicación web sino que también consta de un núcleo de cálculo externo, programado anteriormente, el cual hace las funciones básicas a partir de código sin estructurar ni gestión de los proyectos. Este núcleo está programado por un ingeniero eléctrico y es utilizado por la aplicación web para generar unos resultados determinados.

En este artículo, se mencionan los métodos usados tanto en la metodología como la planificación prevista desde el comienzo del proyecto, el desarrollo donde se llevarán a cabo los objetivos ya mencionados de una forma práctica y con resultados visibles. Este desarrollo se ha subdividido en tres fases: Análisis, Diseño e Implementación, ya que proporciona un cierto grado de orden y estructura respecto al proyecto. Para finalizar se habla de los resultados obtenidos y unas conclusiones del proyecto.

1. ESTADO DEL ARTE

Actualmente, este tipo de proyecto para la automatización de funcionalidades en relación con las catenarias de ferrocarriles, no se ha implementado anteriormente, esto se debe a que es un proyecto de desarrollo interno de empresa.

SIRECA en el pasado, fue un proyecto enfocado de forma errónea, se comenzó su implementación utilizando tecnologías incorrectas, en este caso un framework web escrito en Java llamado Zkoss [3], este framework es recomendado para programadores sin experiencia para crear aplicaciones web sencillas y de un tamaño menor, a causa de esta mala decisión se empezó a desarrollar SIRECA en este contexto resultando un fracaso, ya que se empezaron a ver numerosos fallos que imposibilitaban su desarrollo, como por ejemplo, una mala gestión de la memoria por parte del framework que acumulaba datos sin limpiar en memoria y estaban en uso constante mientras se trabajaba con la interfaz, esto para una aplicación que debe correr en un servidor 24/7 y va a ser usada por numerosos usuarios era un punto negativo muy importante a tener en cuenta, también el framework Zkoss tiene un uso muy rudimentario de las tecnologías web actuales ya que todo lo controla mediante Componentes y no permite escribir código Javascript en el cliente, a causa de esto, limita las posibilidades de diseño e implementación.

Por los motivos expuestos en este apartado, se decidió junto al equipo de SIRECA, rehacer la aplicación desde cero pero con un enfoque diferente, sobretodo en el cuadro tecnológico para que pudiera satisfacer las necesidades de la propia aplicación.

Después de un estudio exhaustivo de las tecnologías actuales se decidió optar por el framework .NET de Microsoft [4], más concretamente ASP.NET en su versión MVC, los motivos principales de esta decisión fueron principalmente la garantía y fiabilidad que aporta una compañía de reputación internacional como es Microsoft, también el gran apoyo por parte de la comunidad y la gran cantidad de documentación al alcance de cualquier persona a través de Internet. Gracias a este estudio y análisis se supo que no se cometerían los errores del pasado por lo que el proyecto comienza con un nuevo enfoque.

En este proyecto, debe quedar claro, que no se limita a la recreación de una aplicación ya existente pero obsoleta, sino, de completar el proyecto hasta el punto definido en el apartado de objetivos, es decir, no solo llegar al punto donde ya estábamos sino superarlo y completar fases que no se habían implementado.

2. OBJETIVOS

SIRECA se divide en fases, concretamente en cuatro, estas fases a grandes rasgos pretenden determinar unas millonadas dentro del proyecto, las fases son:

- Cuaderno de replanteo.
- Dibujado de catenaria.
- Fichas de montaje.
- Fichas de pendolado.

Dentro de este proyecto se han incluido las dos primeras fases, las fichas de montaje y pendolado se han dejado para una futura implementación fuera del alcance de este trabajo de fin de grado.

En la fase cuaderno de replanteo se pretende conseguir generar a partir de una aplicación web un replanteo con datos de un proyecto real y en la fase de dibujado se pretende realizar una previsualización de los planos mediante datos de un proyecto real.

Se han indicado y concretado algunos objetivos o tareas principales de visión general que engloban otras sub-tareas, estos se han cumplido a lo largo del proyecto junto a su desarrollo, es una buena forma de generalizar y promover la organización y gestión de los objetivos más relevantes.

- Lograr los módulos planteados al principio del proyecto y completar los aspectos relacionados con la antigua interfaz web.

- Realizar un estudio e investigación de las tecnologías web actuales y elegir una.
- Definir una arquitectura del software.
- Diseñar e implementar una nueva interfaz web con las nuevas funcionalidades.
- Lograr comunicación y funcionamiento de cálculo entre el núcleo y la interfaz web para implementar la funcionalidad de Cuadernos de replanteo.
- Desarrollar un sistema de pre dibujado y dibujado completo de una catenaria.

3. METODOLOGÍA

Después del estudio de las metodologías más usadas en proyectos reales y descartar las opciones que no se adaptaban correctamente a este trabajo de fin de grado, se ha decidido utilizar la *metodología en cascada o waterfall* [5], pero con detalles o matices de metodologías ágiles como podría ser SCRUMM [6].

Principalmente se usará waterfall para la realización de las tareas a lo largo del proyecto, diferenciando que se aplicarán las *sprint planning meetings* usadas por la metodología SCRUMM, estas reuniones se realizan periódicamente y hay dos tipos, las reuniones de seguimiento y repaso de los requisitos a cumplir con el equipo de trabajo SIRECA dentro de la empresa y las reuniones con el tutor de la universidad para realizar un seguimiento adecuado del proyecto.

A lo que se refiere los objetivos o tareas a alcanzar se realizan individualmente y se adaptarán al método waterfall, el problema principal de este método es su secuencialidad, es decir, se podría dar el caso que en un proyecto con un equipo de trabajo de un tamaño considerable que haya algún tipo de fallo en una fase avanzada, como podría ser la de implementación, esto causaría un error catastrófico en la dinámica del proyecto ya que habría que rehacer la mayoría de tareas. En el caso de SIRECA no se contempla este apunte, ya que al ser diseñado e implementado por una sola persona, no podría darse un fallo así por una mala comunicación con el equipo o que un fallo de programación afectará a otro miembro y desencadene una serie de errores.

Podemos apreciar un ejemplo secuencial en método waterfall con las partes más importantes de un proyecto: análisis, diseño, implementación, pruebas y mantenimiento, podrían haber más dependiendo de los objetivos iniciales.

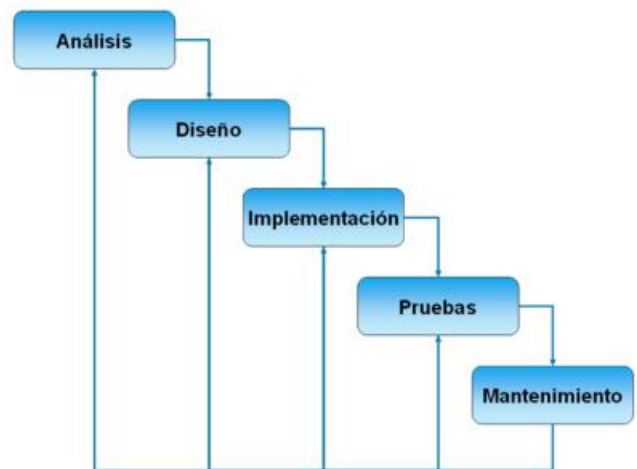


Fig 1. Diagrama de la metodología Waterfall.

4. PLANIFICACIÓN

El diagrama se divide en tareas, y estas tareas se basan en los objetivos propuestos al inicio del proyecto, respecto a estas tareas se ha realizado una estimación de tiempo y fechas según un criterio propio relacionado con la dificultad y la duración de cada tarea.

Las tareas propuestas son de concepto general, esto implica que cada una de ellas se compone por otras muchas sub-tareas a realizar pero que no se han tenido en cuenta en la planificación ya que el objetivo de esta planificación no es describir detalladamente cada tarea concreta, sino generalizar para estimar una duración del proyecto.

Las tareas u objetivos marcados son:

- Estudio e investigación de tecnologías web
- Análisis
- Diseño
 - Interfaz de usuario
 - Arquitectura del software
- Implementación
 - Funcionalidades de la web
 - Comunicación web - núcleo
 - Cuadernos de replanteo
 - Sistema pre-dibujado y dibujado
- Verificación / pruebas / test

Las tareas de análisis, diseño e implementación, se han realizado de una forma no secuencial, esto quiere decir que se han superpuesto entre ellas, de forma que se han realizada de forma paralela ya que, depende del caso, se requería desempeñar tareas de diseño antes que tareas de implementación, esto en el diagrama de Gantt se ve reflejado en la superposición de las tareas respecto al eje temporal.

Se ha llevado un seguimiento del proyecto correcto, intercalando reuniones tanto con el profesor de la universidad como con el equipo de SIRECA en la empresa.

Las fechas propuestas al inicio del proyecto se han cumplido en la medida de lo posible, salvo el último punto del dibujado completo de una catenaria, el cual, no ha podido realizarse dentro de este trabajo de fin de grado por falta de tiempo, pero que se realizará en un futuro.



Fig.2 Arquitectura front-end

5. DESARROLLO

En este apartado del proyecto se comentará todo lo relacionado con el desarrollo, desde principio a fin. Para una mayor estructuración del contenido, se divide el desarrollo en tres secciones: Análisis, Diseño e Implementación. En cada una de estas partes se comentarán todos los aspectos tratados de una forma clara y ordenada.

5.1 ANÁLISIS

A continuación se exponen los conceptos más importantes donde se ha realizado un trabajo de análisis en el proyecto, sobre todo en la arquitectura del software, tanto front-end como back-end.

5.1.1 Front-end

Como todas las aplicaciones web, SIRECA, se divide en dos grandes partes, Front-end y Back-end, la parte front-end o capa de presentación es la parte del software que interactúa con el o los usuarios, esta contiene las vistas HTML que son interpretadas y mostradas al usuario final por un navegador como Google Chrome o Mozilla Firefox y la comunicación con los controladores.

Los lenguajes de programación y sus tecnologías respectivas que se ha utilizado en este proyecto han sido:

- HTML5 para la creación de las vistas de la aplicación web, como ficheros con formato .cshtml relacionados directamente con .NET framework. En estas vistas se ha podido incrustar código C# gracias a la sintaxis Razor [7].
- CSS3 con Bootstrap [8], un framework de estilo web muy usado actualmente, los puntos fuertes son su capacidad Responsive para adaptarse a cualquier tamaño de pantalla o plataforma, totalmente personalizable y de código abierto y contiene funcionalidades jQuery ya utilizables.
- Javascript con jQuery para funcionalidades en el cliente, se ha utilizado una variante del framework llamada jQuery-UI [9], la cual ofrece un mejor aspecto visual en ciertos casos.

5.1.2 Back-end con .NET Framework de Microsoft

La parte back-end o capa de acceso de datos es la parte que contiene toda la lógica de negocio, y la que procesa los datos de entrada del front-end, está formado por los controladores y los modelos que se comunican entre ellos.

La aplicación web SIRECA implementa el patrón de diseño MVC [10] el cual está compuesto por las vistas, los controladores y los modelos. Los modelos son los encargados de extraer información de la base de datos. Los controladores gestionan la información, recogiendo la del modelo para proporcionarla en las vistas, haciendo cualquier acción intermedia necesaria. Finalmente la vista es procesada por un servidor web (en este caso IIS [11]) el cual es atacado por un navegador web de un cliente o usuario que interpreta la página para poder mostrarla por pantalla.

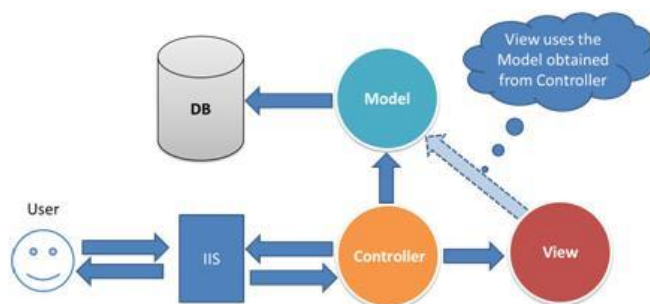


Fig 3. Diagrama estructural del software

Para el desarrollo de la web se ha utilizado el IDE oficial de Microsoft, Visual Studio, el cual permite al desarrollador crear sitios y aplicaciones web gracias a ASP.NET, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos, consolas, etc. En el caso de este proyecto se utilizará el framework .NET 4.5 junto a ASP.NET en su versión MVC junto a Visual Studio 2013 para implementar la aplicación web.

ASP.NET es un framework para aplicaciones web desarrollado por Microsoft. Es usado por programadores y diseñadores para construir sitios web dinámicos.

cos, aplicaciones web y servicios web XML y es la tecnología sucesora de la tecnología Active Server Pages (ASP).

ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework (C/C++, C# y VB).

ASP.NET framework adapta tres modelos de programación distintos:

- Web Forms fue el primero de los tres modelos de programación en existir, y proporciona un gran nivel de abstracción con un modelo de programación familiar basado en eventos y controles.
- ASP.NET MVC se concibió como alternativa a Web Forms y proporciona un modelo de programación basado en el popular patrón de arquitectura MVC. Entre sus principales características destaca su separación más clara entre la lógica de presentación, la lógica de negocio y la lógica de acceso a datos y es el modelo utilizado para SIRECA.
- ASP.NET Web Pages es el más reciente de los tres modelos de programación, y fue creado como respuesta a una creciente demanda de desarrolladores web sin experiencia previa con ASP.NET, proporciona un modelo de programación más simple para programadores novatos.

La aplicación web de SIRECA implementará el modelo de programación de ASP.NET MVC para una mayor gestión y organización de sus recursos, ya que contendrá y manipulará una gran cantidad de datos. Se ejecutará como aplicación de navegador utilizando HTML5, CSS3, Javascript en el cliente y C# en el servidor.

Los motivos principales de la decisión de este framework web, son:

- Microsoft garantiza que la tecnología funcione correctamente y no quede obsoleta.
- Gran cantidad de documentación e información en la red.
- La cantidad de desarrolladores ASP.NET es muy alta, siendo uno de los framework web más usados del mercado.

5.1.3 Núcleo de cálculo

La aplicación web de SIRECA usa un núcleo de cálculo que se encarga de realizar las tareas de cálculo para los replanteos y de dibujado para la previsualización.

Este núcleo está programado en Visual Basic for Applications [12], el cual es un lenguaje de programación orientado a interactuar y cooperar con aplicaciones del

propio Microsoft, como por ejemplo Excel y Access del paquete Office.

En el caso de SIRECA para la fase 1 del, cuaderno de replanteo, trabaja con hojas Excel, las cuales son generadas por la aplicación web y son manipuladas por un módulo de VBA, dando como resultado otra hoja Excel con los resultados obtenidos.

Respecto al dibujado, se llama a un módulo VBA con todos los datos de entrada necesarios proporcionados por el usuario a través de la aplicación web y realiza una previsualización del plano mediante el programa de modelaje AutoCAD, esta previsualización se traduce en una imagen de formato estándar y se muestra al usuario en la web.

5.1.4 Gestión de la configuración

Para el control de versiones del proyecto, se ha utilizado un servidor Subversion [13] con un cliente TortoiseSVN y AnkhSVN.

Subversion o SVN es una herramienta de control de versiones open source basada en un repositorio. Es software libre y está bajo una licencia de Apache. A diferencia de Git y otros sistemas de control de versiones, Subversion no contiene un repositorio local por lo que todos los cambios se realizan directamente en el servidor, el cual está configurado para el control de versiones dentro de la empresa, el cual controla la mayoría de proyectos internos.

Como cliente TortoiseSVN es software libre y se adapta al explorador de Windows para realizar todas las tareas pertinentes con el control de versiones, por otro lado, AnkhSVN es un plug-in para Visual Studio que permite realizar las acciones de control de versiones desde el mismo IDE.

5.1.5 Documentación del código

En este proyecto la documentación del código se ha tomado como una parte importante, se ha utilizado Doxygen como herramienta para la automatización de la documentación, en resumen, la función de Doxygen es traducir y clasificar a través de etiquetas específicas en el código los comentarios del desarrollador, todo esto se procesa y se genera una serie de vistas HTML en la cual se puede ver toda la información referente al proyecto y así facilitando su uso a futuros desarrolladores.

5.1.6 Servidor de test y producción

Actualmente, SIRECA consta de dos servidores virtuales proporcionados por Sener, servidor de test y de producción, uno de ellos se utiliza para realizar pruebas por usuarios ajenos al proyecto, como son otros ingenieros de la propia empresa, efectúan acciones con la aplicación web

para comprobar la usabilidad de la interfaz, además del correcto funcionamiento de la aplicación y sus funcionalidades.

El servidor de producción no está en marcha aun, se pretende utilizar cuando SIRECA pase a ser un software estable y fiable, para empezarse a usar en proyectos reales y aprovechar sus resultados.

5.2 DISEÑO

En este apartado se explican los principales conceptos de diseño en el proyecto, el diseño de la base de datos y de la interfaz de usuario.

5.2.1 Modelo de la base de datos

La base de datos utilizada fue la misma que ya había creada, el motivo es que estos datos seguían siendo los mismos y ya estaban bien estructurados. Estos datos se preservan en una base de datos MySQL, para adaptarla a la aplicación web en ASP.NET se tuvo que instalar otros componentes como el MySQL conector ODBC y el Entity Framework.

El modelo de la base de datos está preparado para mantener una gran cantidad de datos, aun así, una base de datos relacional como puede ser MySQL es suficiente para el proyecto. El modelo está formado por una serie de tablas, de las cuales las más importantes son:

Normalmente, cuando se realiza una aplicación web con ASP.NET framework, es recomendable usar SQL Server como base de datos ya que es totalmente compatible y no necesita de software de terceros, en este caso al ya tener la base de datos en MySQL se estimó que el tiempo de migración era innecesario y se optó por el conector ODBC de MySQL.

Entity Framework es un ORM [14], consigue realizar un mapeo de las tablas de la base de datos, a clases dentro del código, convertir los atributos de las tablas en propiedades de estas clases, por lo que esto ofrece una gran ventaja a la hora de desarrollar la aplicación, además .NET con Entity Framework también ofrece la posibilidad de crear los controladores y las vistas respecto a un modelo (tabla de la BBDD) automáticamente, referenciando unas rutinas CRUD [15] (Create, Read, Update, Delete) y permitiendo modificarlo a gusto del desarrollador.

5.2.2 Interfaz de usuario (UI)

La interfaz de usuario sigue un estilo conservador, minimalista e intuitivo, respetando el color corporativo de Senner (azul) como sello de identidad, en el apéndice A se observa alguna de las vistas como ejemplo de la aplicación y en la figura 4 un diseño genérico de la interfaz.

SIRECA tiene una interfaz sencilla de utilizar, pero enfocada a personas del sector, es decir, gente calificada y preparada para proyectos de este tipo y pertenecientes a la empresa.

Las partes principales del *layout* de la UI son:

- Header: se encuentra por un lado el título del proyecto, mientras que por otro, en el caso de haber introducido las credenciales, se mostrará el proyecto activo en el caso de que lo haya, e indicará que no hay ningún proyecto activo en el caso de que no lo haya. También se muestra el nombre de usuario logeado.
- Footer: simplemente contiene información sobre la empresa así como un link a la página web oficial.
- Sidebar: contiene el acceso a todas las opciones disponibles: usuarios, cables, macizos, postes, catenarias, proyectos, cuadernos de replanteo, predibujado. Permanecerá visible durante todo el uso de la aplicación salvo en la pantalla de login.
- Principal: todo el contenido de la aplicación se recarga en este contenedor.

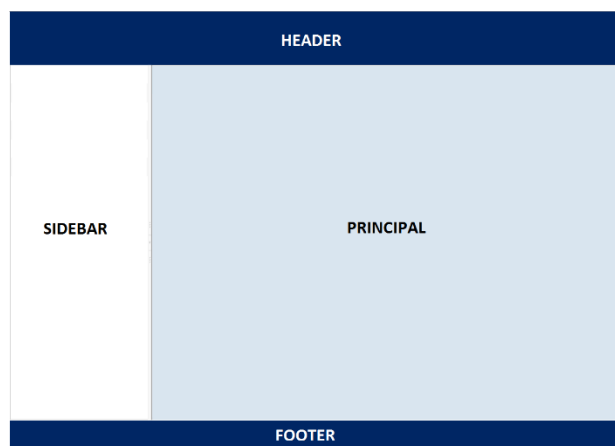


Fig 4. Diseño de la interfaz genérico

Las categorías o apartados que podemos encontrar en la aplicación web son: Usuarios, cables, macizos y postes que permite la creación, modificación y eliminación de estos y se almacenan para ser seleccionados en el siguiente apartado de catenarias, en este tenemos una lista de catenarias disponibles y la posibilidad de crear nuevas, este módulo es donde más datos se almacenan junto a proyectos, si editamos una catenaria podremos gestionar toda su información, lo mismo pasa en proyectos pero estos están relacionados directamente con una catenaria, es decir, los datos entre ellos son independientes. La siguiente opción es cuadernos de replanteo donde podemos encontrar la información de las versiones y revisiones, y las funcionalidades que podemos usar, como calcular un nuevo replanteo,

importar uno que ya se ha creado, entregar la versión actual o descargar una plantilla de datos de entrada. La última es la opción del menú de predibujado, en ella podremos seleccionar un predibujado existente y modificarlo, eliminarlo o crear uno nuevo desde su comienzo, podemos ver algunos ejemplos de la interfaz para las opciones mencionadas anteriormente en el apéndice A.

La pantalla principal también contiene un menú desplegable para acceder a la información del usuario o hacer log-out. En la esquina inferior derecha de la pantalla se ha puesto el número de versión de ese momento.

5.3 IMPLEMENTACIÓN

En esta última parte del desarrollo, se expone los apartados más importantes relacionados con el proyecto y la forma en la que se han realizado, podemos saber la estructura de la aplicación, el funcionamiento de los cuadernos de replanteo y del predibujado.

5.3.1 Estructuración

El proyecto se implementa en un entorno de desarrollo con Visual Studio, se estructura de forma que una solución puede contener varios proyectos que se relacionen entre sí, en este caso, la solución creada es SIRECA la cual contiene tres proyectos, SIRECA Web, SIRECA Database y SIRECA Test.

SIRECA Web: contiene toda la estructura principal del proyecto y su contenido, sus partes más importantes son las vistas separadas por secciones que son ficheros cshtml adaptados para poder insertar código C# en la vista, los controladores que gestionan los datos y funcionalidades para controlar tareas como crear, eliminar o guardar datos en la base de datos, el directorio Content que almacena todos los ficheros de estilo CSS y las imágenes utilizadas en la web, la carpeta Scripts que guarda los ficheros javascript y jQuery para ser accedidos por las vistas, los Resources que contiene una clase de variables constantes como por ejemplo las variables de entorno del sistema necesarias para el proyecto y un fichero resources que contiene todas las cadenas de texto de la aplicación facilitando en un futuro la posibilidad de agregar nuevos idiomas, los Model-View que gestionan datos que no son guardados directamente en base de datos pero que son necesarios y los CoreService que son las clases que interactúan con el núcleo de cálculo como el predibujado y los replanteos

SIRECA Database se crea para mantener una mayor modularidad en la aplicación, de forma que solo guarda los modelos creados a partir del Entity Framework y estos son accedidos por los controladores de SIRECA Web para modificar la base de datos.

SIRECA Test está orientado a los test unitarios que se utilizan para comprobar el correcto funcionamiento de las funcionalidades.

5.3.2 Cuaderno de Replanteo

En el apartado de cuaderno de replanteo la aplicación permite visualizar un listado de todas las versiones y revisiones del proyecto, con los datos de cada una de estas revisiones, crear una nueva revisión, entregar versión así como eliminar la última versión siempre y cuando no sea la número 1, y descargar la plantilla a rellenar para el cálculo.

La lógica de negocio principal en los cuadernos de replanteo es que cada versión de replanteo contiene una lista de revisiones de esa versión del replanteo y es lo que vemos reflejados en la interfaz web como vemos en la figura 5.

REVISIÓN	USUARIO	FECHA	TIPO	ESTADO	TAMAÑO	ACCIONES
1	admin	17/05/2016 11:53:59	Calculado	Completado	1251 Kbytes	[Icons]
2	admin	20/05/2016 10:28:05	Calculado	Completado	1251 Kbytes	[Icons]
3	admin	20/05/2016 11:24:47	Calculado	Completado	155 Kbytes	[Icons]

REVISIÓN	USUARIO	FECHA	TIPO	ESTADO	TAMAÑO	ACCIONES
1	admin	22/05/2016 11:56:35	Importado	Completado	174 Kbytes	[Icons]

Fig 5. Vista de cuaderno de replanteo

Replanteo Versión: Representa a una versión del cuaderno de replanteo. Dado que las versiones son un simple contenedor de revisiones, esta entidad solamente guarda el número de versión, el identificador del proyecto y un booleano que indica si puede o no ser eliminada.

Replanteo Revisión: Representa a una revisión dentro de una versión del cuaderno de replanteo. La entidad permite conocer el número de revisión (dentro de la versión), el tipo (calculado, importado o recalculado), la fecha de creación, el tamaño del fichero asociado y el estado actual (calculada, calculada con warnings, con errores, o procesando).

El sistema de ficheros se encuentra una carpeta llamada "projects" con todos los ficheros asociados a los proyectos de la herramienta. Para cada proyecto, hay una carpeta cuyo nombre es el identificador del proyecto en la base de datos. Además, cada carpeta de proyecto tendrá las siguientes subcarpetas (una por cada fase de la herramienta): "calcula-replanteo", "dibujo-replanteo", "fichas-montaje" y "fichas-pendolado". En estas subcarpetas se guardan los ficheros asociados a cada fase de la herramienta, todo esto se muestra en la figura 6.

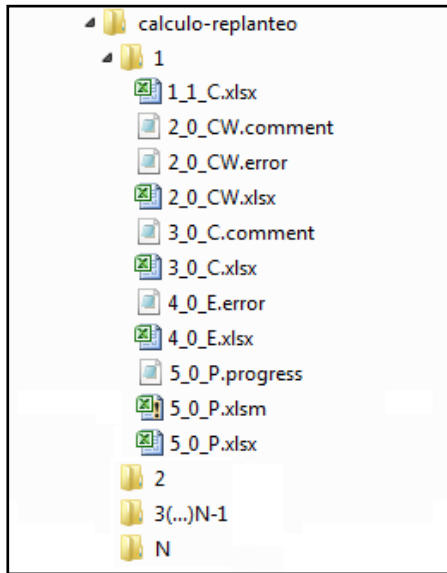


Fig 6. Estructura de directorios

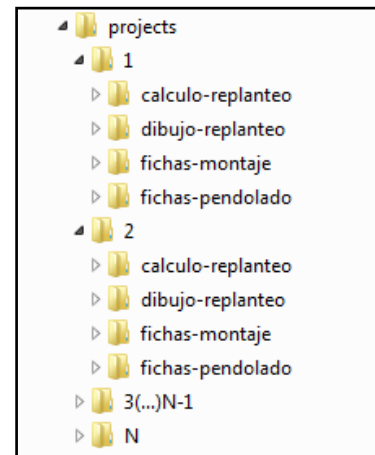


Fig 7. Estructura de archivos de un replanteo

A la hora de trabajar con el sistema de ficheros, la interfaz web administra las versiones y revisiones del cuaderno de replanteo dentro de la carpeta llamada “cuaderno-replanteo”, situada en la carpeta de cada proyecto, en esta se generan directorios numerados con el número de la versión del replanteo, dentro de cada directorio de versión, estos archivos se clasifican según su nombre, siguiendo el siguiente formato:

<Revisión>_<Tipo>_<Estado>.xlsx

Donde:

- <Revisión>: Número de revisión
- <Tipo>: Tipo de revisión. Puede tomar los siguientes valores:
 - 0: Calculado
 - 1: Importado
 - 2: Recalculado
- <Estado>: Estado actual de la revisión. Puede tomar los siguientes valores:
 - P: Procesando
 - C: Completado
 - E: Error fatal
 - CW: Completado con warnings

La opción de cálculo de replanteo nos abrirá una ventana nueva e inhabilitará el contenido de fondo, en esta nueva ventana debemos seleccionar el archivo Excel de entrada con los datos pertinentes, este fichero debe seguir el estándar de la plantilla que es accesible en esta pantalla, también habrá que insertar datos como el PK (punto kilométrico) inicial y final, y opcionalmente unas notas. Cuando empiece a calcular se creará un thread paralelo para que podamos seguir usando la interfaz web mientras se realiza el cálculo en segundo plano. Una vez finalizado aparecerá como una nueva revisión y habrá creado los ficheros output en su directorio correspondiente y genera el fichero de notas. Desde de la interfaz podremos descargar la revisión del replanteo, mirar las notas o eliminar la revisión.

Si descargamos una revisión y vemos el resultado del replanteo podremos observar todos los datos identificativos de esa catenaria ya calculados en la worksheet Replanteo, en la figura 8 podemos observar un fragmento diminuto de los datos de salida ya que el archivo original contiene alrededor de 3500 filas de datos.

8	9	Nº de pylone	PK (m)	Puerta (m)	Implantación (m)	Rayon (m)	Devers (mm)	Desavement 1 (m)	Desavement 2 (m)	Hautour (m)	Pendulage 1	Pendulage 2
10	11	0-1A	0=371,6		1,7	882,69	56,64	0,20		5,50		
				31,50							SK8a	54A31,5
12	13	0-3F	0=403,1		1,7	539,87	92,62	0,25	0,65	5,50		
				27,00							SK9k	559I2
14	15	0-5F	0=430,1		1,7	405,03	123,45	-0,15	0,25	5,50		
				27,00							559w2	559v2
16	17	0-7F	0=457,1		1,7	398,96	125,32	-0,15	0,25	5,50		
				27,00							559I2	SK9j
18	19	0-9AF	0=484,1		1,7	398,96	125,32	-0,15	0,25	5,50		
				27,00							SK9b	54A27
20	21	0-11A	0=511,1		1,7	398,96	125,32	0,20		5,50		
				31,50								C5n31,5
22		0-13	0=542,6		1,7	398,96	125,32	0,20		5,50		

Fig 8. Fragmento del resultado de un replanteo

La opción de importar replanteo nos abrirá una ventana nueva e inhabilitará el contenido de fondo, en esta nueva ventana debemos seleccionar el archivo Excel con el resultado de un replanteo y las notas para la revisión. En este caso no realiza ningún cálculo, simplemente importa el fichero subido y lo incorpora como el resultado de un cálculo, lo guarda en su respectivo directorio y genera el fichero de notas.

5.3.3 Predibujado

El objetivo de esta fase de pre dibujado, es en esencia, conseguir configurar una serie de datos para procesarlos a través de un módulo de dibujado el cual ejecutará la aplicación de AutoCAD para automatizar el proceso de dibujado de un solo tramo de la catenaria completa, gracias a los datos asociados en la aplicación web. En la figura 10 se muestra el ejemplo de un trazado de un dibujado ya realizado sin utilizar la aplicación de SIRECA, es un ejemplo concreto, ya que con la aplicación de SIRECA podremos añadir o no, todas las entidades que hayan disponibles realizándolo de forma automática.

Cada pre dibujado contendrá un conjunto de entidades gráficas, estas entidades tendrán sus propios atributos los cuales identificaran: la posición, el color, la imagen entre otras características, estas servirán para colocar la entidad en su lugar correspondiente en el pre dibujado.

La lógica estará dividida en dos clases, Predibujado y Entidad Grafica, cada predibujado contiene un listado de entidades gráficas, y cada entidad grafica tiene todas las características necesarias para ser localizada en el pre dibujo.

En la vista de predibujados podemos encontrar todos los predibujados disponibles y las posibilidades de crear, modificar o eliminar. Si modificamos un predibujado veremos dos columnas de entidades, la primera columna contiene todas las entidades graficas disponibles, la cual esta directamente relacionada con un directorio del proyecto el cual es totalmente escalable y ampliable dinámicamente, el usuario elegirá la entidad que quiere añadir a su predibujado y la arrastrará hasta la segunda columna, donde se mostrarán todas las entidades que actualmente están seleccionadas, al arrastrar la entidad, aparece una ventana emergente donde nos pedirá que insertemos los datos referentes a esa entidad.

Todo esto se puede observar en la figura 9, como apunte adicional en la imagen se puede observar como las entidades son unos cuadrados con una cruz a modo de entidad genérica, esto se debe a que los iconos aún se han seleccionado, en el momento que estén disponibles será tan sencillo como añadir las imágenes a los directorios correspondientes.



Fig 9. Vista de generación de un predibujado.

Toda esta información estará preservada en la base de datos para que cuando el usuario genere el predibujado el núcleo de dibujado pueda recoger la información necesaria desde la tabla correspondiente.

Una vez finalice el dibujado la aplicación generará dos ficheros asociados a ese predibujado, una imagen en formato PNG y otra en formato DWG, estas dos se guardarán en disco y estarán directamente relacionadas con un predibujado, en la figura 10 podemos observar un resultado final de generar un predibujado.

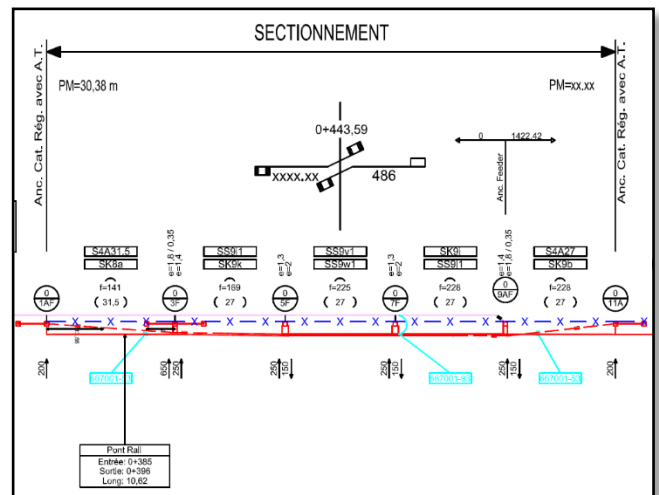


Fig 10. Ejemplo de predibujado

6. RESULTADOS

Los resultados respecto a SIRECA, han sido generalmente satisfactorios, aun así al proyecto aún le queda mucho por delante ya que se considera un proyecto enorme, en cuanto a este trabajo de final de grado, se han llegado a cumplir prácticamente todos los objetivos planteados al principio, se ha llegado al punto de partida rehaciendo la aplicación web de una forma más óptima y escalable usando otras tecnologías más viables que las usadas anteriormente, se ha

implementado la fase 1 del desarrollo, los cuadernos de replanteo, pudiéndose ejecutar y obtener el resultado del cálculo remotamente contra un servidor virtual interno de la empresa, de forma que se ha realizado una comunicación de aplicación web con núcleo de cálculo exitosa, en cuanto a la fase 2 y al predibujado, se ha dejado preparada la aplicación web para introducir todos los datos por parte del usuario, de forma que cuando finalice el desarrollo del núcleo de dibujado por la otra parte del equipo de SIRECA se podrá implementar sin problemas.

El objetivo planteado al principio del dibujado de la catenaria ha tenido que ser aplazado por falta de tiempo, por lo que se realizara fuera de este trabajo de fin de grado, este consistía en dibujar no solo un tramo sino en dibujar toda una catenaria a partir de los datos de la aplicación.

7. CONCLUSIONES

Para concluir el trabajo se presentan las conclusiones extraídas de estos 5 meses de desarrollo.

Se ha sido capaz de desarrollar una herramienta que realice cálculos de replanteos y dibujados de catenarias automáticamente sobre un proyecto real de la empresa, actualmente SIRECA no ha desarrollado todo su potencial, y en un futuro se espera que pueda ser independiente y que facilite el trabajo a muchos ingenieros, ya que ese el objetivo principal del proyecto. Gracias a esta aplicación, cuando esté finalizada, otros ingenieros podrán ahorrar semanas de trabajo ya que SIRECA podrá realizar esas tareas en unos minutos, en esencia, es una vista al futuro y una inversión muy acertada.

Se considera que se ha llegado a un buen punto y un gran avance en el proyecto SIRECA dentro de este trabajo de fin de grado, cumpliendo casi todos los objetivos planteados por ambiciosos que pareciesen al comienzo.

El trabajo futuro que se espera es optimizar aplicación web y núcleo junto con desarrollar nuevas funcionalidades y características como el dibujado completo de una catenaria o las fichas de pendolado y montaje.

AGRADECIMIENTOS

Me gustaría agradecer a mi empresa Sener, donde he desarrollado el proyecto y me han ofrecido la posibilidad de participar en un proyecto real e integrarme en un entorno de desarrollo laboral, a mi equipo de SIRECA, Meritxell Cusidó, Edgar Calderón, y sobre todo Héctor García por ayudarme y asesorarme a lo largo del proyecto, también agradecer a mi tutor de la universidad Ramón Martí por guiarme en todo el proceso de documentación y resolver mis dudas.

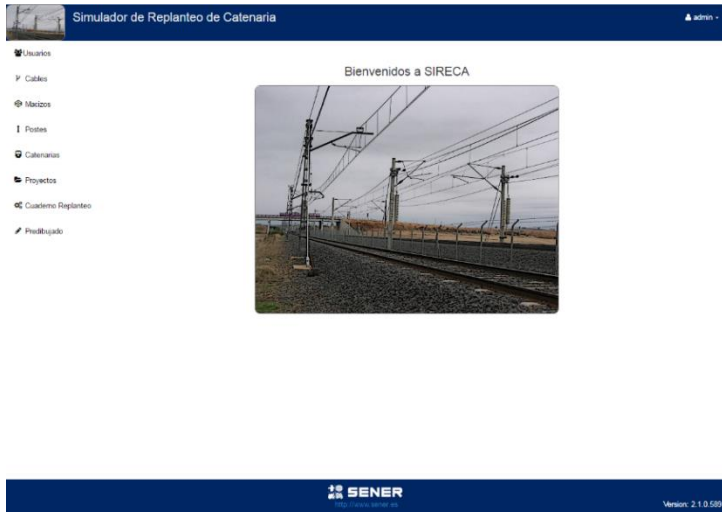
BIBLIOGRAFÍA

- [1] Sener Ingeniería y Sistemas. 2016 S.A. 6 de junio. <http://www.sener.es>.
- [2] Catenaria, Enciclopedia.us.es. 6 de junio [http://enciclopedia.us.es/index.php/Catenaria_\(ferrocarril\)](http://enciclopedia.us.es/index.php/Catenaria_(ferrocarril)).
- [3] Zkoss, Java web framework. 8 de junio <https://www.zkoss.org/>.
- [4] .NET, Framework de Microsoft. 8 de junio [https://msdn.microsoft.com/eses/library/zw4w595w\(v=vs.110\).aspx](https://msdn.microsoft.com/eses/library/zw4w595w(v=vs.110).aspx)
- [5] Metodología waterfall o cascada. 8 de junio <https://highscalability.wordpress.com/2010/06/29/diferencias-entre-las-metodologias-waterfall-y-agil-diferencias-between-waterfall-and-agile-methodologies/>
- [6] SCRUMM metodología ágil. 8 de junio. <https://proyectosagiles.org/que-es-scrum/>
- [7] Razor, ASP.NET Syntax. 12 de junio. http://www.w3schools.com/aspnet/razor_syntax.asp
- [8] Bootstrap, CSS framework responsive, 23 de junio. <http://getbootstrap.com/css/>
- [9] jQuery-UI, Javascript framework, 23 de junio. <http://jqueryui.com/>
- [10] MVC Pattern (Model, View, Controller), 23 de junio. <https://msdn.microsoft.com/en-us/library/ff649643.aspx>
- [11] IIS Server (Internet Information Services), 23 de junio. <http://www.iis.net/>
- [12] Visual Basic for Applications, 19 de junio. <https://msdn.microsoft.com/en-us/library/office/gg264383.aspx>
- [13] Subversion version control, 23 de junio. <https://subversion.apache.org/>
- [14] ORM (Object Relational Mapping), 23 de junio. [https://msdn.microsoft.com/es-es/library/bb399182\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/bb399182(v=vs.100).aspx)
- [15] Funcionalidades CRUD, 23 de junio. <http://www.asp.net/mvc/overview/getting-started/getting-started-with-ef-using-mvc/implementing-basic-crud-functionality-with-the-entity-framework-in-asp-net-mvc-application>

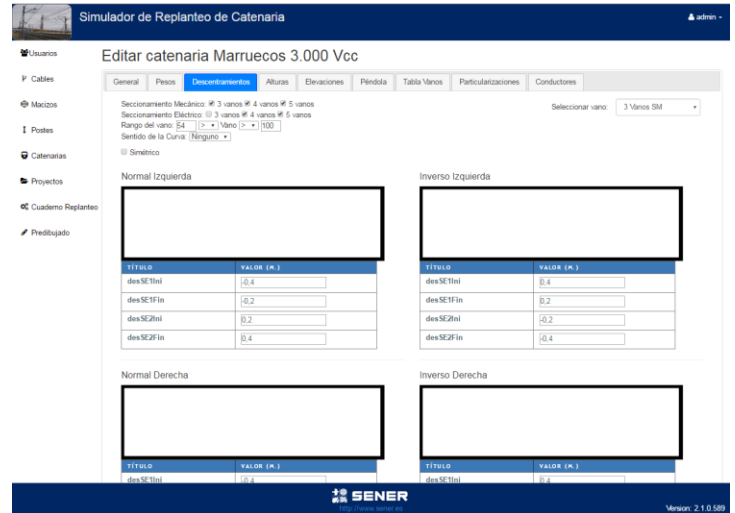
APÈNDIX

A. INTERFAZ DE USUARIO

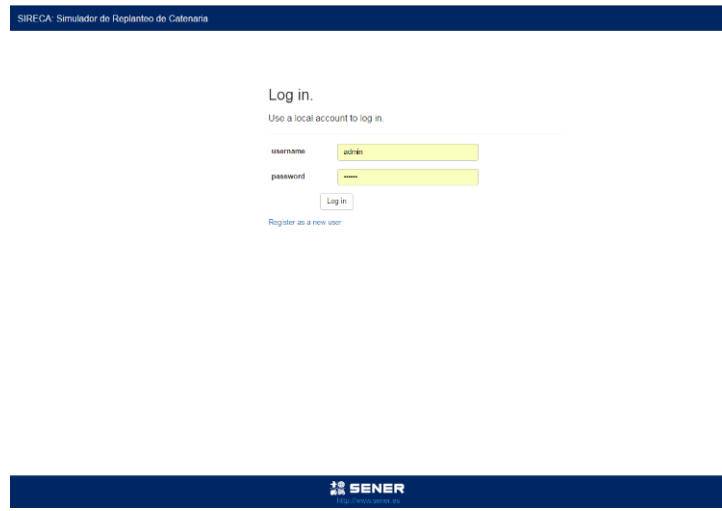
A.1 Pantalla de bienvenida



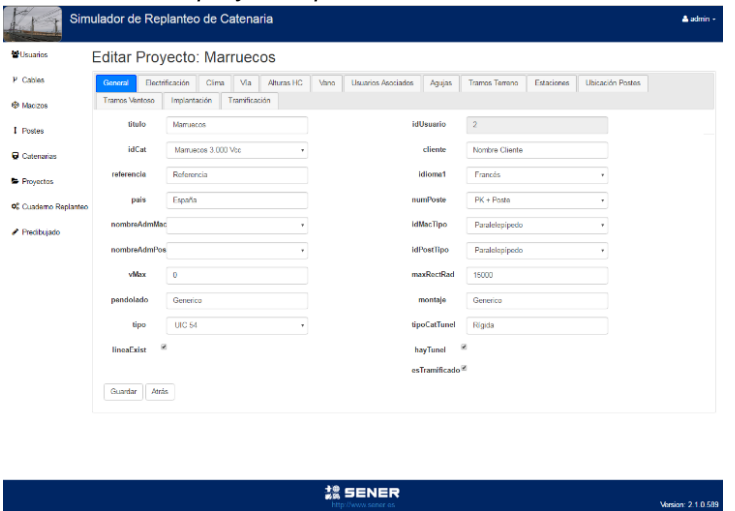
A.2 Editar una catenaria, pestaña “Descentramientos”



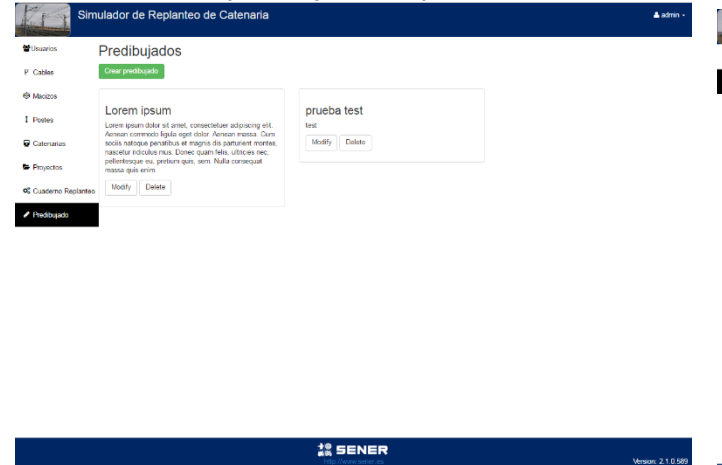
A.3 Pantalla de log-in



A.4 Editar un proyecto, pestaña “General”



A.5 Lista de predibujados disponibles



A.6 Lista de cables disponibles

