
This is the **published version** of the bachelor thesis:

Umasi Yomona, Kerly Sandivel; Estravis Nieto, Sara, dir. WebScraping : Automatización en el dispatching de herramienta de tiqueting. 2021. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/257836>

under the terms of the  license

Web Scraping: Automatización en el dispatching de herramienta de tiqueting

Kerly Sandivel Umasi Yomona

Resumen— El objetivo principal de todas las empresas es la optimización de sus procesos y buen uso de sus recursos enfocados a la productividad, no obstante, al contar con una gran variación de servicios, se encuentran en la necesidad de organizar y tener toda esta información centralizada, sobre todo si hablamos de incidencias dentro de la empresa; es por eso que las herramientas de tiqueting juegan un papel importante para realizar un seguimiento de incidencias, y esto no sería posible sin la intervención de un *dispatcher* o persona que balancea la carga de trabajo entre los diferentes técnicos comprometidos con el sistema. Gracias a las nuevas tecnologías sobre minería de datos con implementaciones en la experiencia del cliente final, tales como un CRM, podemos mejorar el proceso de toma de decisiones en una empresa. El objetivo final del proyecto es hacer uso de una herramienta de tiqueting y mediante una tecnología como WebScraping poder recopilar, analizar, presentar datos de diferentes fuentes, identificar incidencias recurrentes y/o diferentes intervenciones, de esta manera, automatizar el *dispatching* prescindiendo del proceso manual y así disminuir el tiempo final.

Palabras clave— Customer Relationship Management (CRM), Enterprise Risk Management (ERM), Bussiness Intelligence (BI), Big data, Web Scraping, Tiqueting, Service Level Agreement (SLA)

Abstract— The main objective of all companies is the optimization of their processes and good use of their resources focused on productivity, however, having a great variation of services, they find themselves in the need to organize and have all this centralized information, especially if we talk about incidents within the company; That is why ticketing tools play an important role in tracking incidents, and this would not be possible without the intervention of a dispatcher or person who balances the workload between the different technicians committed to the system. Thanks to new technologies on data mining with implementations in the end customer experience, such as a CRM, we can improve the decision-making process in a company. The final objective of the project is to make use of a ticketing tool and through a BI technology to be able to collect, analyze, present data from different sources, identify recurring incidents and / or different interventions, in this way, automate dispatching regardless of the manual process and thus decrease the final time.

Index Terms— Customer Relationship Management (CRM), Enterprise Risk Management (ERM), Bussiness Intelligence (BI), Big data, Web Scraping, Tiqueting, Service Level Agreement (SLA)



1 INTRODUCCIÓN

Actualmente y desde siempre ha sido muy valorada la eficacia dentro de una empresa, así como la de los mismos trabajadores, como de las herramientas desarrolladas para ejecutar tareas monótonas o nuevas. El desarrollo y la innovación tecnológica se encuentra en constante cambio, y son las ventajas de estas nuevas tecnologías acompañadas de la incorporación de CRM lo que nos permitirá mejorar la estrategia para así poder asumir nuevos retos.

El cliente al cual va dirigido este proyecto pertenece al sector de retail, una multinacional que cuenta con más de 630 tiendas repartidas alrededor de España, 11 plataformas logísticas y con una plantilla de más de 17000 empleados. La empresa se encuentra en constante expansión, así como sus nuevas tecnologías ya sea en sus tiendas, almacenes o centros logísticos.

La empresa cuenta con un sistema de tiqueting el cual les permite monitorizar incidencias que puedan afectar a la

producción ya sean alertas en servicios internos o escalados de proveedores externos. La asignación de estos tiquets se realiza bajo un *dispatching*, el cual viene bajo la responsabilidad de una o diferentes personas, las cuales tienen que ser capaces de deducir el problema con los datos recopilados del sistema, así también asignarles una prioridad o el tipo de afectación o impacto que tiene dentro de la empresa.

Un sistema de *tiqueting* nos permite conocer incidencias dentro del entorno, así poder hacer seguimiento y gestionarlo junto a otras peticiones de servicios ya sean solicitudes hechas desde los propios usuario o incidencias que se generan automáticamente después de encontrar una vulnerabilidad en el sistema [1].

Este software genera un ticket por cada solicitud que crea el propio usuario o la propia herramienta de monitorización.

Es posible automatizar el proceso de hacer *dispatching*, ya que gran cantidad de incidencias dentro de la empresa, suelen ser repetitivas o tener palabras claves que puedan identificar su afectación y calcular el impacto que puede tener.

- E-mail de contacte: Kerly.umasi@gmail.com
- Menció realizada: Tecnològies de la Informació
- Trabajo tutorizado por: Sara Estravís Nieto (DEIC)
- Curso 2021/22

Así como el software es importante para una ágil asignación de tiquets, la revisión de estadísticas de seguimiento de productividad por cada trabajador también es cruciales para poder ser analizados, de esta manera dar a conocer puntos de inflexibilidad los cuales nos darán un enfoque a los problemas recurrentes, de esta manera poder trabajar sobre ellos dándoles una solución definitiva, evitando las soluciones temporales.

Para la implementación y desarrollo de este trabajo, se hará uso de diferentes tecnologías, metodologías de organización y el uso de algoritmos para poder mejorar nuestro entorno y asignar respuestas según sea la necesidad del caso lo requiera.

1.1 Motivación

Actualmente no existe un visualizador o plataforma dentro de la herramienta de tiqueting donde se pueda visualizar la carga de trabajo por cada técnico o realizar automáticamente el dispatching de tiquets, los cuales pueden llegar a haber entre 80-100 tiquets diarios.

El departamento cuenta con diferentes equipos que se especializan en temas concretos, por lo que a veces un mismo tiquet se tiene que gestionar entre diferentes equipos, lo que conlleva más tiempo de gestión.

Después de analizar el tiempo invertido en el servicio de dispatching, el cliente se ve en la necesidad de implementar una herramienta que le permita gestionar estas incidencias de una manera automatizada.

La forma en que se obtendrán estos datos desde la herramienta de tiqueting, será mediante la técnica de *Web Scraping*, la cual consiste en la extracción automatizada de contenido desde la red, este método puede usarse en casos que se requiera extraer poca información, pero la mejor forma de aprovecharlo es para grandes cantidades de información.

El trabajo consiste entonces, en la selección adecuada de datos dentro de un sitio web que sería la herramienta de tiqueting, la asignación automática a un técnico basándose en la visualización de la carga de trabajo que tiene.

Haciendo uso de este proyecto, nuestros resultados mejorarían el *feedback* de cada usuario ya que el tiempo de resolución Service Level Agreement (SLA) [2] se podría reducir. El soporte, la asignación y gestión de incidencias también serían más eficaces, por ende, mejoraríamos tanto el soporte como el *feedback* de cada usuario o de cada tienda, se pueden reducir las incidencias abiertas por usuarios o incidencias dentro de una tienda, un almacén, un centro logístico, etc. Además, es motivación personal el aprender sobre nuevas tecnologías en vanguardia dentro del sector e-commerce, ya sea ERM, CRM, BI, Web Scraping, etc.

1.2 Objetivos

El objetivo principal es implementar una mejora mediante un análisis de Web Scraping en la web de tiqueting utilizada para la repartición de carga de trabajo.

Otros objetivos del trabajo son:

- Además de una gestión empresarial para una óptima gestión de recursos, se externalizarán los datos obtenidos a otra aplicación y según estas métricas se calculará la productividad media, carga de trabajo diario a tiempo real, creación de dashboards diferenciando servicios afectados y ver el porcentaje de afectación según cada servicio.
- Otro objetivo del Proyecto es mejorar el tiempo de eficacia de cada trabajador, al estar enfocado para la resolución de incidencias en la empresa, así como en tiendas o Ware House (WH), los resultados finales para el equipo de CRM ayudarían a anticipar los problemas, focalizándose en los patrones de las incidencias, reduciéndolos y disminuyendo el tiempo de afectación para el cliente final.
- Estandarizar los KPIs definidos para un flujo habitual de incidencias y su resolución.
- Focalizarnos en puntos críticos que causan mayor impacto o incidencias repetitivas.
- Utilizar y poner en práctica tecnologías aprendidas durante la carrera.
- Agilizar, optimizar un problema de un caso real.

1.3 Planificación

Este proyecto se realizará en fases, desglosando diferentes tareas que se tendrán que realizar cada semana y alguna de ellas con dependencia.

Por cada semana se irán añadiendo o modificando cada tarea. Para ello se usará un diagrama de GANTT.

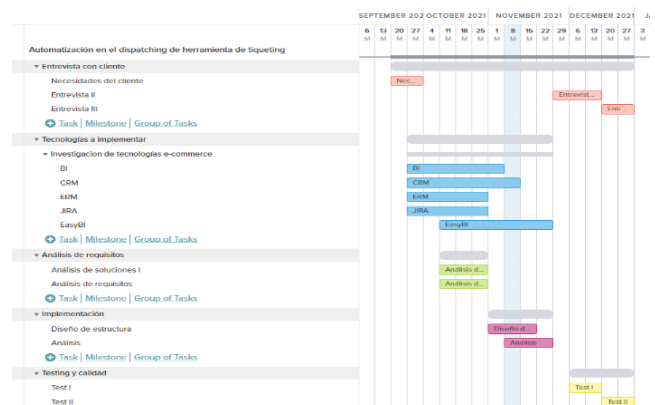


Imagen 1 - Planificación con diagrama de Gantt

2 METODOLOGÍA

2.1 Metodología Agile

Durante el proceso del TFG iré basándome en la metodología Scrum que se basa en principios ágiles [3] para poder desarrollar, testear y volver a modificar los requisitos según necesidad del cliente o según necesidad del proyecto. Principalmente, se hará énfasis en seguir el principio de “Sencillo pero suficiente” y ser “people-oriented” y “Communication-centered”. Como bien se denomina ligero, es más adecuado para el desarrollo de pequeños proyectos [4].

Este método consiste en la retroalimentación o feedback constante para obtener una información nueva basándonos en cada etapa de la realidad, obteniendo así una perspectiva distinta a la que se tenía en la fase anterior. Con este método no solo agilizaremos, sino mejoraremos el rumbo de cada elemento así poder mejorar el producto final.

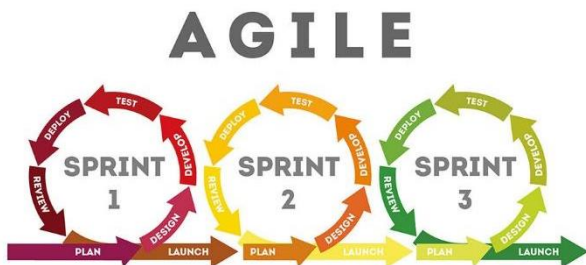


Imagen 2 - Estructura de metodología Agile

3 TECNOLOGÍAS UTILIZADAS

3.1 Implementación de Business Intelligence (BI)

La inteligencia empresarial (BI) es un Decision Support System (DSS) basado en datos que combina la recopilación de datos, el almacenamiento de datos y la gestión del conocimiento con el análisis para proporcionar información para el proceso de decisión.[5].

En la actualidad podemos encontrar diferentes tecnologías con las cuales podemos hacer minería de datos, tales como; MicroStrategy, Microsoft Power BI, EasyBI, etc., Con cada una de estas herramientas podemos hacer un análisis, recopilando información para el uso de estadísticas. Con el uso o implementación de una herramienta basada en inteligencia artificial, podremos predecir comportamientos o detectar patrones dentro de diferentes casuísticas.

3.2 IT4You

Herramienta interna utilizada para la creación de tiquets, la cual recopila información extraídas por otra herramienta de monitorización llamada Nagios [6]. Dentro de IT4You es donde nos centraremos a recopilar información, analizando cada *tag* dentro del *html* de la página, analizaremos y asignaremos un nombre final del técnico responsable a solucionar dicha incidencia. Basaremos el proyecto en dos casos de usos, los cuales se mencionarán más adelante.

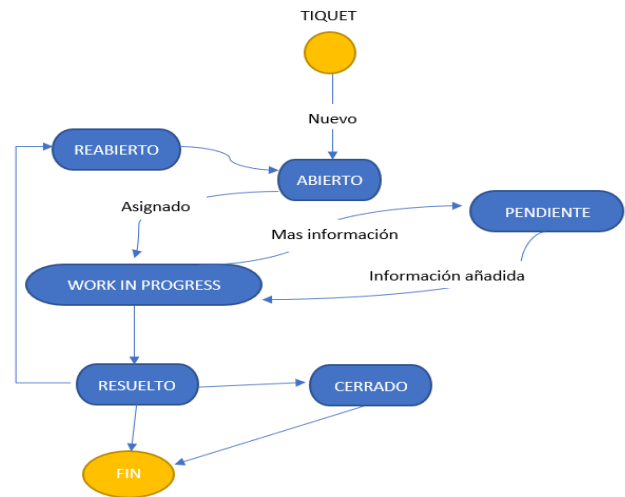


Imagen 3 – Diagrama de flujo para los estados de un tiquet

3.3 CRM

Las estrategias CRM o ERM[7] juntamente con el área de I+D trabajan en la mejora de productos y servicios para cumplir las expectativas de los clientes.

La nueva dinámica del mercado es saber escuchar y entender lo que el cliente desea, ya que esto es la clave para ir amoldando los procesos de negocio a lo que se espera de ellas, desarrollando productos/servicios innovadores, ajustando procesos, mejorando prácticas y evolucionando continuamente de manera alineada con las expectativas de los clientes.

3.4 Big data/ minería de datos

Es fundamental que las empresas creen estrategias para la extracción y modelado de datos, pero la consecuencia de esta estrategia es la “Infoxicación”, ya que a veces se obtienen **tantos** datos que se hace casi imposible seleccionarla o hacer uso de ella con efectividad.

Para este proyecto no se usará una cantidad masiva de datos o metadatos, pero haremos uso de una tecnología llamada “Web Scraping” la cual es muy usada dentro de la minería de datos.

3.5 GitHub

Todo el contenido del proyecto será guardado en la herramienta de Github. Para la fase del desarrollo es imprescindible tener copias de seguridad y diferentes versiones, en caso de Rollback.

3.6 Web Scraping

La técnica de Web Scraping [8] consiste en la extracción de datos, información, imágenes, documentos (entre otros) directamente desde un navegador web. Su realización puede ejecutarla un usuario de manera manual o un bot, aunque lo más frecuente es que sea de manera automatizada. Tiene variadas aplicaciones y es posible implementarlo en distintas plataformas y con distintos procesos, es decir, no hay un método único de realizar Web Scraping.

Se implementará un Web Scraping utilizando como lenguaje de programación NodeJS, el cual es una combinación de **JavaScript y Node**. Por un lado Javascript nos permitirá interactuar y/o manipular los recursos del servidor con un servicio web y **NodeJS** nos facilitará el entorno de ejecución para que nuestra aplicación se pueda ejecutar en el servidor y en el cliente.

Librerías necesarias para la implementación:

- **Cheerio** - Implementación veloz y flexible del core jQuery, diseñado específicamente para la creación de un servidor. Contribuye al paso de información DOM.
- **Axios** - Promise basada en HTTP-client, para el navegador y Node.js. Nos permitirá hacer *fetch* de las páginas mediante HTTP requests.
- **Express** - Aplicación framework flexible, la cual nos permite usar un conjunto de herramientas para la recopilación de datos desde una página o aplicaciones móviles.
- **Cors** - Mecanismo que utiliza cabeceras HTTP para obtener permisos y acceder a recursos seleccionados de un servidor.
- **Puppeteer** - Api de alto nivel para controlar un navegador Headless creado, el cual no es un navegador "real" y nos permitirá para este proyecto automatizar procesos webs, extraer contenido web.
- **SuperAgent** - Al igual que Axios, Superagent es otro cliente HTTP que soporta Promises y sintaxis como *async/await*.

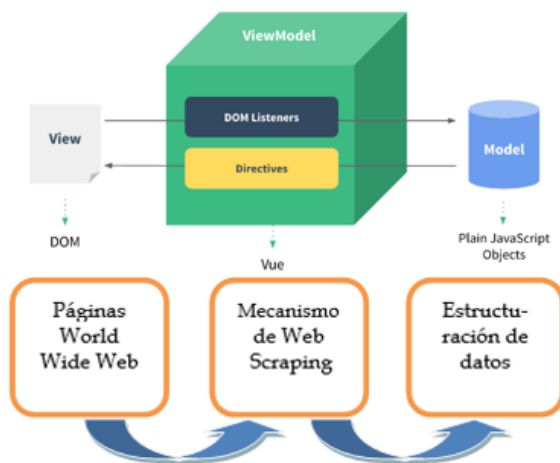


Imagen 4 - Tecnología Web Scraping

3.7 KPI

La información relevante con la que se trabaja cada tiquet se fundamenta con indicadores llamados KPIs (Key Process Indicators)[9].

En este proyecto se trabajará en una técnica que busca obtener la información necesaria para la aplicación de diferentes indicadores de rendimiento (KPI) los cuales pueden ser utilizados para mejorar los diferentes procesos

presentes en las diferentes etapas de una incidencia, facilitando así la aplicación de los KPI propuestos.

Los **KPI** se caracterizan por poseer un número de cualidades, estas pueden ser definidas mediante métricas **SMART** [9], a continuación se define cada término:

- **S:** Específico, un KPI debe ser claro y enfocado en el objetivo que se desea medir.
- **M:** Medible, un KPI se debe poder expresar en términos numéricos.
- **A:** Alcanzable, los objetivos de un KPI deben ser razonables y realizables.
- **R:** Realista o relevante, un KPI es directamente atingente al trabajo realizado en el proyecto.
- **T:** Temporales, un KPI debe permitir su monitoreo dado un período de tiempo.



Imagen 5 - KPI de carga de trabajo

Carga de trabajo por departamento

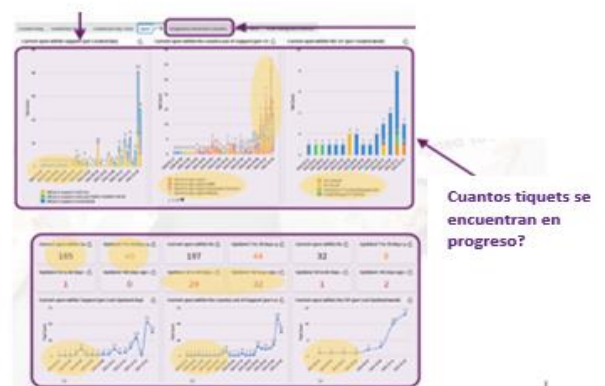


Imagen 6 - KPI para departamento o estado de tiquets

Los KPIs definidos para este proyecto:

- [S] Reducir cantidad de tiquets totales.
- [M] Mejorando en tiempos medios y máximo de respuesta ante una incidencia o petición.
- [A] Gracias al ahorro de tiempo en la resolución de tiquet.
- [R] Grado de satisfacción del usuario final o de toda una delegación.
- [T] En un tiempo gradual.

- [S] Reducir tiempo de dispatching.
- [M] Hasta quitar el 100% de involucración del dispatcher.
- [A] Automatizando el dispatching de tiquets
- [R] Mejorando el porcentaje de incidencias finalizadas fuera de plazo.
- [T] Viendo mejoras en el primer semestre.

- [S] Disminuir carga de trabajo por técnico.
- [M] Obteniendo una productividad en ascenso.
- [A] Mejorando tiempos de SLA por cada tiquet.
- [R] Eliminando tiquets repetitivos o reabiertos.
- [T] Tiempo gradual.

- [S] Cambiar a una gestión 24/7.
- [M] Sin involucración de técnico.
- [A] Mejorando el proceso de automatización y de auto implementación.
- [R] Añadiendo gradualmente diferentes *tags* para abarcar más servicios dentro del Web Scraping implementado.
- [T] Largo plazo y de manera gradual.

Otros KPI's objetivos:

- Sobrecostes eliminados por cada día de incidencia crítica abierta.
- Reducción en días de impacto de afectación en tiendas o almacenes.
- Mejora de tasa de producción o de servicio(Throughput).

4 IMPLEMENTACIÓN

Para esta fase nos basaremos en el patrón Modelo-Vista-Controlador(MVC)[10], por un lado se obtiene todo organizado para futuras implementaciones, así actualizar, modificar o depurar el código.

Por otro lado se divide el código *frontend* y *backend* en componentes separados.

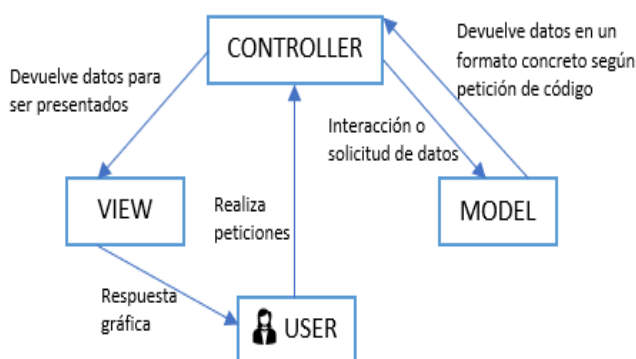


Imagen 7 - Metodología de Modelo-Vista-Controlador

En esta parte del proyecto, se han recreado diferentes tiquets de casos de uso, similares a los que se utilizan dentro de la compañía, con estos, se realizan las pruebas de Web Scraping.

Se crea una web en un host gratuito a la que se puede acceder desde internet.

Simulación web de tiqueting.
<https://ticketingauto.000webhostapp.com/>

Contenido de cada tiquet:

- Detalles del usuario que abre la incidencia.
- Sistema de monitoring que abre automáticamente la incidencia.
- Prioridad de la incidencia.
- Servicio afectado.
- Detalles de la incidencia, donde es posible encontrar sistema(s) afectado(s).

4.1 Casos de uso

Se especifican 3 casos de uso muy habituales en la empresa, la implementación de la herramienta se enfocará en estos:

4.1.1 Caso de uso 1 - Tiquet nuevo

Tiquet que ha llegado recientemente a la cola de incidencias, el cual ha sido creado por medio de la intervención de usuario, técnicos proveedores o herramienta de monitorización.

Características del nuevo tiquet:

- Tiquet pendiente de ser asignado a un técnico.
- Asignación de prioridad.
- Asignación de servicio afectado.

Para esta casuística, se implementa Web Scraping, obteniendo por medio de librerías de NodeJS, el tag HTML donde introduciremos de manera automática el técnico que menor carga de trabajo tenga o de manera manual.

Nos basaremos en la arquitectura de la Imagen 7, donde veremos el flujo dentro del Controlador como en la Imagen 4.

Procedimiento:

1. Web Scraping a través de un pipe lee contenido HTML de tiquet 1.
2. Dentro de los tags comparamos si el tag es igual al "#id=technician" (Tag indicativo para el contenido en formato texto dentro del atributo *input* de la web).
3. Comparamos si el #id se encuentra dentro de la clase de estilo CSS definida (información que se puede recopilar analizando el código fuente de la web).
4. Inserción; el nombre del técnico que nos haya devuelto la BBDD (obtenido según carga de trabajo o inserción manual) será la que automáticamente nos aparezca en el campo obtenido en el paso 2 y 3.
5. Tenemos 2 maneras de guardar los datos en la BBDD.
Manual; el dispatcher decide si la elección es la correcta y en tal caso guarda el contenido o lo asigna a otro técnico.

Automático; según el algoritmo de carga de trabajo lo considera, se guardará de manera automatizada, sin la supervisión del dispatcher.

El diagrama de flujo utilizado para resolver el caso de uso 1 puede verse en la Imagen 8.

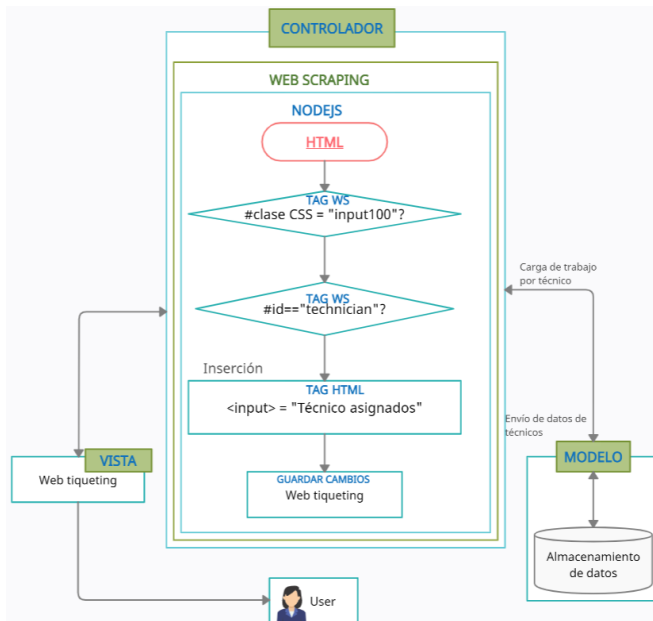


Imagen 8 - Diagrama de flujo para Caso 1

4.1.2 Caso de uso 2 – Carga de trabajo por técnico

Este proceso nos servirá como un Dashboard que el dispatcher o las personas que lo solicitan podrán revisar, ya sea para una vista general de la carga actual o para asignar tiquets nuevos al técnico que menos carga tenga.

Para este caso de uso nos basaremos en los datos obtenidos con el Web Scraping y lo usaremos para tenerlos de forma visual.

Características del tiquet:

- Tiquet creado y con técnico asignado.
- Implementación con dependencia de la obtención de datos del caso 1.
- Creación de Dashboard basándose en la carga de trabajo por cada técnico. Dato que se obtendrá de la BBDD.

Procedimiento:

1. Siguiendo el procedimiento del caso de uso 1 y después de guardar la variable del nombre del técnico en la BBDD.
2. Se solicitará a la base de datos el nombre de los técnicos guardados.
3. Comparamos el tag #technician obtenido de Web Scraping con los nombres de los técnicos guardados.
4. En caso de coincidir aumentaremos el contador para la carga de trabajo.
5. Guardamos los cambios.

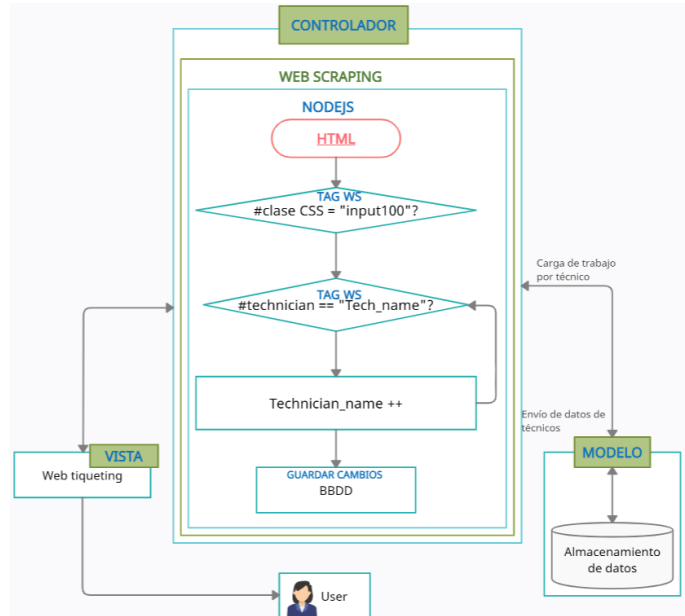


Imagen 9 - Diagrama de flujo para Caso 2

4.1.3 Caso de uso 3 - Tiquet en fase de resolución

Tiquet que actualmente está siendo gestionado por un técnico y posiblemente por varios equipos. Es posible que para la gestión del tiquet haya sido enviado a la revisión de otro equipo. Es por esta razón que cuando detectemos el retorno del tiquet, el dispatcher revisa el historial del tiquet y lo asigna nuevamente al técnico que previamente lo estaba gestionando.

Para este caso dependeremos de la correcta obtención de tags y comparación con los nombres de los técnicos, de tal manera detectar el técnico que se encuentra en el proceso de resolución del tiquet.

Características del tiquet:

- Tiquet creado y con técnico asignado en la primera instancia.
- Implementación con dependencia de la obtención de datos del caso 1.
- Tiquet que ha sido gestionado por un segundo técnico de un segundo departamento o equipo.
- Pendiente de volver a ser asignado.

Procedimiento:

1. Seguiremos mismo flujo de la imagen 8 con la diferencia de que en lugar de buscar en solo una clase CSS, buscaremos en una segunda clase CSS.
2. Por cada clase encontrada, Web Scraping comparará el dato obtenido dentro del tag #id == technician con los técnicos registrados en la base de datos.
3. En caso de encontrar una coincidencia, el atributo texto HTML dentro del tiquet se renombrará con el mismo nombre del técnico asignado en la primera instancia.

5 RESULTADOS

5.1 Resultados caso de uso 1

Después de la implementación de Web Scraping y de la obtención de los datos dentro de los tags HTML definidos, obtenemos los siguientes resultados:

- En primer lugar se obtienen los datos extraídos de la URL del tiquet.
- En segundo lugar pasamos a la fase de procesamiento de Web Scraping.
- En tercer lugar se realiza la ejecución de la función `Asignar_tecnico()`.
- Y en último lugar, en caso de aprobar la asignación, se guardan los datos.

Tiquet 20211202

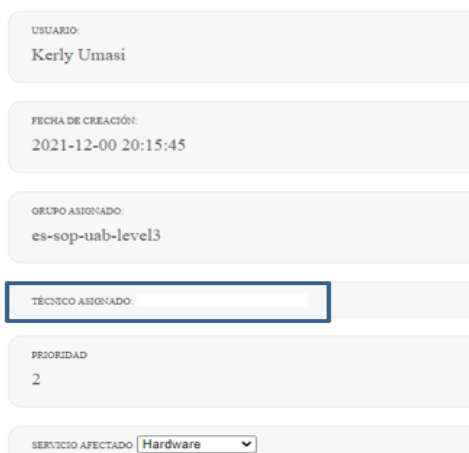


Imagen 9 - Tiquet sin técnico asignado

Web Scraping
`Asignar_tecnico()`

Tiquet 20211202

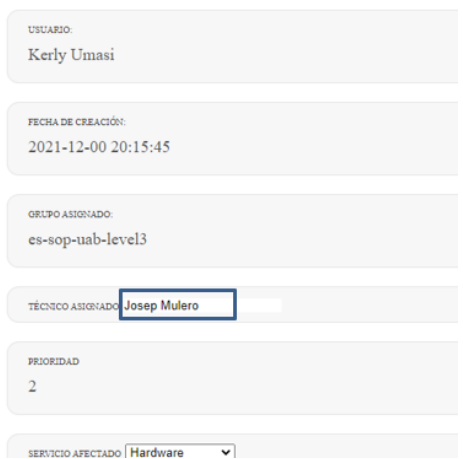


Imagen 10 - Automatización de asignación de técnico

Observaciones del cliente:

La preferencia del cliente para este apartado es optar por la asignación automática sin previa revisión antes de guardar el nombre del técnico asignado, ya que comenta que si simultáneamente se logra asignar varios tiquets, después comprobaría la carga de trabajo, y basándose en esta podría reasignar tiquets en caso de que los resultados no sean correctamente equitativos con la repartición de carga.

4.2.1 Resultados caso de uso 2

Posterior a la recopilación de datos en el caso 1 se observa:

- Resultado de la carga de trabajo por cada técnico.
- Contador de prioridades activas totales.

Carga de trabajo

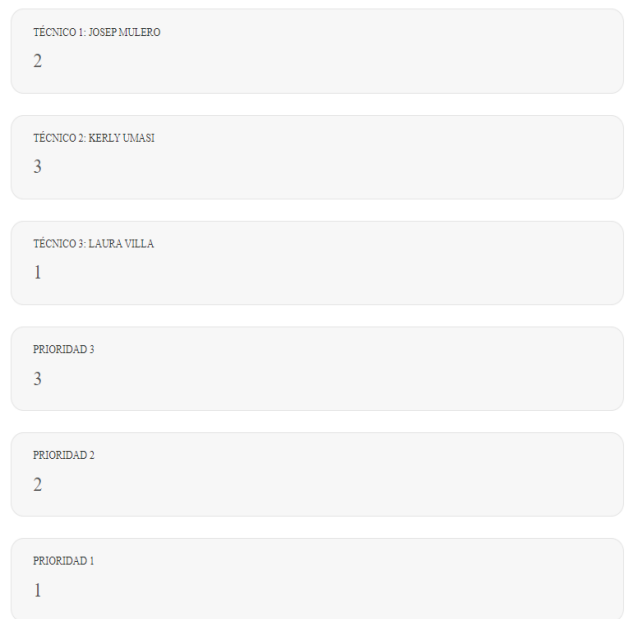


Imagen 2 - Tiquets totales por cada técnico y cuantificación de prioridades

Observaciones del cliente:

El cliente comenta que este apartado lo revisaría solo para corroborar que las asignaciones entre técnicos sean correctas. Comenta también que el tiempo ahorrado es bastante notorio.

El cambio que añadiría sería implementar una vista para técnicos asignados y otra para prioridades de tiquets.

Además preferiría en lugar de tener la cantidad total, mantendría un listado de todos los tiquets, y al momento

de pasar el cursor por encima, tener una breve descripción del tiquet y poderlo reasignar fácilmente al técnico que considere.

4.2.2 Resultados caso de uso 3

Dentro de la implementación del *Web Scraping* en el código, se ha definido extraer los resultados que obtenemos dentro del tag llamado "Técnico", el cual nos brinda la información del técnico que actualmente está trabajando dentro del tiquet, este histórico se lee dentro del HTML completo, se compara si el nombre del técnico existe dentro de la base de datos y finalmente se le asigna este mismo tiquet para continuar la incidencia.

Este apartado no se ha podido completar en una versión final, por lo que se agrega al apartado de implementaciones futuras.

Observaciones del cliente:

Se comenta que este apartado sería beneficioso al igual que el caso 1 ya que sería ahorro de tiempo. El cliente añadiría a la implementación el hecho de poder relacionar un técnico con una especialidad y que la asignación se base primeramente en la especialidad y después en la carga de trabajo.

6 CONCLUSIONES

El objetivo principal de este trabajo era la implementación de una automatización mediante un análisis de Web Scraping en la web de tiqueting utilizada para la repartición de carga de trabajo ha sido finalizado con buenos resultados funcionales.

Después de comparar lenguajes de programación como Python[11], o tecnologías como Scrapy[12], la elección de NodeJs ha sido por consecuencia la más acertada, debido a su gran versatilidad y las diferentes librerías que se pueden importar, esto nos ha permitido evitar ciertas implementaciones irrelevantes para pocos tiquets, tal como una base de datos o un servidor web exclusivo.

Debido a la planificación, experiencia o tiempo, algunos objetivos citados en el proyecto, no se han sido concluidos en su totalidad. Tales como mejora en el estilo de cada página web, Dashboard definitivo para cliente, comparación de carga de trabajo entre técnicos. Estos puntos se agregan dentro del "Desarrollo futuro y ampliaciones".

Actualmente, por motivos de seguridad no se ha podido implementar la solución descrita en este proyecto en la red interna del cliente, sin embargo las pruebas hechas con plantillas *html* similares a los tiquets utilizados en situaciones reales han sido muy satisfactorias y el cliente está muy satisfecho porque considera que puede aportar mucho valor y reducción de tiempo y satisfacción tanto en clientes, como en tiendas o almacenes.

La metodología Agile ha permitido que en cada paso avanzado y/o finalizado se pueda ir constantemente evaluando. En cada *Sprint* se organiza una reunión con el cliente para mostrarle los resultados obtenidos y posibles mejoras, pero la homogeneidad en la metodología entre el cliente y el proyecto ha sido correcta.

Como una de las conclusiones importantes que se podría destacar, sería que a nivel laboral y educacional, es que sin una buena organización no es posible llegar a los diferentes Sprints o concluir correctamente las tareas predefinidas dentro de una estimación de tiempo.

Por otro lado, el hecho de que una empresa tenga correctamente estandarizados unas fases o desarrollo de tareas, por más pequeñas que sean, son tan relevantes como las tareas habituales ya que algún fallo es posible corregirlo a tiempo, de esta manera aprovechar recursos y optimizar una correcta productividad.

7 DESARROLLO FUTURO Y AMPLIACIONES

En un futuro se plantea la posibilidad de Testing dentro de la intranet corporativa.

Siendo un proyecto que abiertamente se pueden considerar diferentes nuevas funcionalidades, se podría ampliar de la siguiente manera.

- El desarrollo del caso de uso 3 ha quedado implementada parcialmente, ya que por falta de experiencia o tiempo no se ha podido finalizar automáticamente para que la intervención del dispatcher sea totalmente nula.
- No se ha dado prioridad a las CSS del estilo en cada página de test, pero en una implementación futura se considera seriamente este punto.
- Se realizó un primer diseño conceptual de una base de datos, pero se descartó ese apartado al no tener una carga masiva de datos. Es punto a considerar en un desarrollo futuro cuando se pueda testear dentro de la empresa. De esta manera realizar un API en backend que pueda interactuar directamente con las peticiones HTTP desde JSON hacia la base de datos MySQL de forma más versátil y posiblemente más tolerante a fallos con las librerías de Node.js.

8 AGRADECIMIENTOS

Si hay algo que tengo que agradecer fuertemente en este mundo, es a la gran persona que con su apoyo incondicional siempre ha sabido guiarme, aconsejarme, estar conmigo en las buenas, malas y peores; mi madre. Sin ella, o sin mi gran JanP. no hubiese podido llegar a donde estoy, son mis pilares de vida. Mi padre, uno de mis modelos a seguir por su perseverancia y coraje que

siempre muestra. Estoy segura de que ellos ven mis esfuerzos y se sienten orgullosos de mí y de verme crecer en cada pasito, ya que no ha sido fácil el compaginar la carrera con una vida laboral.

Voy a dedicar un párrafo a mi pareja ya que él siempre está dispuesto a apoyarme en todo, aunque se trate de una locura. Una gran persona que sabe la dedicación que una carrera conlleva, su apoyo, sus consejos, sus correcciones, sus risas.

Agradezco a todos mis compañeros y compañeras, a los que seguimos manteniendo el contacto, como a los que no; ya que he aprendido y compartido mucho o menos con cada uno de ellos.

Por ultimo y no menos importante, mi más sincero agradecimiento a mi tutora Sara Estravís, que ha sido paciente conmigo y ha sabido ser versátil con algunos cambios dentro del proyecto, sabiendo dar un nuevo enfoque.

Gracias.

BIBLIOGRAFÍA

- [1] Icedmed (2018) "Herramienta de ticketing: que son y como usarlas para mejorar el CX".
Disponible en:
<https://www.esic.edu/rethink/comercial-y-ventas/herramientas-de-ticketing-que-son-y-como-usarlas-para-mejorar-el-cx>
- [2] Wu, L., & Buyya, R. (2012). "Service level agreement (SLA) in utility computing systems."
Disponible en:
<https://arxiv.org/ftp/arxiv/papers/1010/1010.2881.pdf>
- [3] Arturs R., & Solvita B. (2017) "Method for Adaptation and Implementation of Agile Project Management Methodology."
Disponible en:
<https://reader.elsevier.com/reader/sd/pii/S187705091730056X?token=2111244259732D19A9F6EBF20234A084B083FCCE969F9CD1E95AF19C48813F609EDBF811915B2739F11D324B5CB6699&originRegion=eu-west-1&originCreation=20211010190133>
- [4] Gaurav K., & Pra, K. (2012) "Impact of Agile Methodology on Software Development Process"
Disponible en:
https://www.researchgate.net/publication/255707851_Impact_of_Agile_Methodology_on_Software_Development_Process
- [5] Solomon Negash (2020) "Business Intelligence"
Disponible en:
https://link.springer.com/chapter/10.1007/978-3-540-48716-6_9
- [6] Nagios. Consulta desde la web oficial.
Disponible en:
<https://www.nagios.com/products/nagios-xi/#:~:text=Nagios%20XI%20is%20available%20in%20two%20different%20editions%3A,aid%20in%20large-scale%20configuration%2C%20forecasting%2C%20and%20scheduled%20reporting.>
- [7] Mark S. B (2016) "What is Enterprise risk management"
Disponible en:
https://erm.ncsu.edu/az/erm/i/chan/library/What_is_Enterprise_Risk_Management.pdf
- [8] Anand V. S., Kedar G. P., Shweta A. G. (2018) "An Overview On Web Scraping Techniques And Tools"

Disponible en:

<http://www.ijfrcsce.org/index.php/ijfrcsce/article/view/1529/1529>

- [9] Harold Kerzner Ph.D (2013), "Project management metrics, KPIs, and dashboards a Guide to Measuring and Monitoring Project Performance"

Disponible en:

<http://library.wbi.ac.id/repository/216.pdf>

- [10] Yenisleidy F. R. & Yanette D. G. (2012) "Patrón Modelo-Vista-Controlador"

Disponible en:

<https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15/10>

- [11] Web oficial Python.

Disponible en:

<https://www.python.org/doc/essays/blurb/>

- [12] Dimitrios Kouzis L. (2016) "Learning Scrapy"

Disponible en:

https://books.google.es/books?hl=es&lr=&id=EF8dDAAAQBAI&oi=fnd&pg=PP1&dq=scrapy&ots=KHT9BzTZun&sig=92S60V7tpdmc_l6M9znYCr0IQNE&redir_esc=y#v=onepage&q=scrapy&f=false