



**UNIVERSIDAD
DE ANTIOQUIA**

**DISEÑO DE EXPERIENCIA Y DESARROLLO DE
APLICACIÓN WEB DE TABLEROS DE CONTROL
EN LA DIRECCIÓN DE VIDA Y RENTAS**

Autor

Cristian Camilo Bedoya Zapata

Universidad de Antioquia

Facultad de Ingeniería

Medellín, Colombia

2020



Diseño de experiencia y desarrollo de aplicación web de tableros de control en la dirección
de vida y rentas

Cristian Camilo Bedoya Zapata

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:
Ingeniero de Sistemas

Asesores:

Carlos Mauricio Duque Restrepo, Ingeniero de Sistemas

Juan Esteban Carvajal Barrios, Ingeniero de Sistemas

Universidad de Antioquia

Facultad de Ingeniería

Medellín, Colombia

2020.

Resumen

La dirección de vida y rentas tiene numerosos miembros que están divididos en tres equipos, los cuales se encargan de desarrollar proyectos nuevos, mantener y mejorar las aplicaciones actuales y brindar soporte a estas. Por lo tanto es necesario conocer el estado de las tareas de cada uno de estos equipos para poder tomar acciones de mejora y poder mantener una comunicación coherente con los clientes. Razón por la cual se propuso implementar un tablero de control unificado, en el cual participen los clientes directos de éste en su proceso de definición y creación, en el que además se pueda tener en tiempo real el estado de los sprints pasados, el sprint actual y los incidentes de cada equipo de trabajo, pero que además sea visible para todos los miembros de la dirección de manera que se logren generar ideas y accionables desde cualquier rol.

Introducción

En la dirección de seguros de vida y rentas se necesita entender el comportamiento de trabajo dentro de sus diversos equipos: Proyectos, mejora continua y operaciones. Estos equipos se encargan de desarrollar nuevos proyectos e ideas, mejorar y mantener las aplicaciones existentes, además brindar soporte y mantener la operación diaria. Entonces, adquirir este conocimiento ayudará a analizar los procesos actuales para tomar prontas acciones de mejora en caso de ser necesario y tratar de predecir comportamientos futuros para tomar decisiones acorde a los datos hallados.

Para lograr esto se realizan tableros de control en los que se concentrará la información de todos los desarrollos, sprints, e incidentes en el tiempo, de manera que se pueda filtrar y comparar la información día a día para obtener una visión global de lo que ocurre dentro de la dirección.

Por lo tanto, con el fin de crear tableros de control que brinden una experiencia positiva, se hace una exploración inicial en la que se buscan métodos, técnicas y software que permita una correcta identificación de la necesidad de los usuarios, expectativas, gustos y demás factores que afecten en la utilización de los tableros de control. Para posteriormente definir los datos a usar y prototipar los tipos de gráficos necesarios, de manera que se pueda presentar la información requerida de forma concisa, ésto aplicando una metodología iterativa como lo es Design Thinking, para así lograr integrar a los clientes durante todo el proceso de creación y llegar a un producto final adecuado a sus necesidades.

Luego de esto continúa una exploración tecnológica para determinar la forma en que se implementarán tales tableros, teniendo en cuenta buenas prácticas en desarrollo software, además de costo de implementación y así, posteriormente, crear una aplicación web funcional o la implementación en herramientas ya existentes, pero basada en el mejor diseño de interfaz de

usuario, experiencia de usuario y diseño de software. Adicionalmente, tal implementación abarca la integración con datos productivos.

Objetivo General

Diseñar la experiencia de usuario y desarrollar una aplicación web de tableros de control para la dirección de vida y rentas.

Objetivos Específicos

- Identificar métodos para encontrar las necesidades de los usuarios que usarán el tablero de control.
- Identificar métodos para encontrar el perfil de los usuarios que usarán el tablero de control.
- Identificar arquitecturas de referencia front-end, frameworks, librerías o tecnologías que permitan implementar los tableros de control.
- Diseñar y crear prototipos de interfaces de usuario que se adapten al perfil y necesidades de los usuarios que usaran el tablero de control.
- Obtener datos reales de Jira y libros de Excel para ser proporcionados como insumo a la aplicación de tableros de control.
- Desarrollar una aplicación donde se pueda visualizar la información requerida por la dirección de vida y rentas.

Marco Teórico

Diseño de experiencia de usuario:

Se trata de mejorar la experiencia de los usuarios mediante la creación de productos o servicios que resulten placenteros de usar. Esto significa averiguar exactamente cuál es el problema que se está atacando y cómo piensa, siente e interactúa la persona.

Design Thinking:

Es una metodología que integra experticias en diseño, ciencias sociales, ingeniería y negocios, con un enfoque en el usuario final para generar productos, servicios o sistemas innovadores brindando una mejora continua con cada iteración. Esto con el fin de crear un ambiente interactivo que promueva el aprendizaje a través del rápido prototipado.

Esta metodología consta de 5 etapas:

1. **Empatizar:** Se busca entender la necesidad desde el punto de vista de los usuarios, teniendo en cuenta quienes son, sus experiencias previas, gustos, conocimientos e incertidumbres.
2. **Definir:** Se define el problema a resolver basados en los datos recolectados de la etapa 1.

3. **Idear:** Se generan ideas sobre una posible solución al problema planteado en la etapa 2.
4. **Prototipar:** Se crean prototipos (Pueden ser de baja o alta fidelidad) en el menor tiempo posible basados en la mejor idea de la etapa 3.
5. **Probar:** Se entrega el prototipo a los usuarios para experimentación y se obtiene retroalimentación de su parte.

Prototipo de baja fidelidad:

Son normalmente prototipos basados en papel, los cuales no permiten una interacción profunda por parte del usuario. Sin embargo, estos prototipos son más rápidos de crear y permiten visualizar más fácilmente diseños alternativos para brindar una mejora. Además, una ventaja de este tipo de prototipo es que quien interactúe con él se sentirá más cómodo al momento de sugerir cambios.

Prototipo de alta fidelidad:

Son prototipos hechos mediante software, que normalmente deben permitir interacción realista por parte del usuario (Uso de gestos, escritura...), por lo tanto, estos prototipos deben ser una representación lo más cercana posible a la interfaz de usuario final. Adicionalmente, estos prototipos son útiles para recolectar información real de uso humano, por ejemplo, medir el tiempo para realizar cierta tarea.

Front-End:

Es la parte de un programa o dispositivo a la que un usuario puede acceder directamente e interactuar. Por lo que son todas las tecnologías de diseño y desarrollo web que funcionan en el navegador y que se encargan de la interactividad con los usuarios.

Scrum:

Es un proceso de gestión que reduce la complejidad en el desarrollo de productos para satisfacer las necesidades de los clientes. Aquí tanto los dueños del producto como los equipos de desarrollo trabajan juntos alrededor de requisitos y tecnologías para entregar productos funcionando de manera incremental. Con este marco de trabajo también viene el término Sprint, el cual data del tiempo que pasa desde la planeación hasta la entrega de valor.

Scrum Master:

El Scrum Master es quien vela por la correcta utilización de Scrum, la remoción de impedimentos y asiste al equipo para que logre su mayor nivel de desempeño posible. Puede ser considerado como un coach o un facilitador encargado de acompañar al equipo de desarrollo. Le corresponde iniciar y facilitar las ceremonias de Scrum, ejecutar los eventos y utilizar los instrumentos correctamente. También, debe atender al Dueño del producto creando transparencia sobre el progreso del equipo.

Historia de usuario:

Una historia de usuario es una representación de un requisito de software escrito en una o dos frases utilizando el lenguaje común del interesado. Brindan

descripciones claras y concisas de la funcionalidad en términos del valor que aporta al usuario final del producto y permiten administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir mucho tiempo para administrarlos. Las historias de usuario brindan descripciones claras y concisas de la funcionalidad en términos del valor que aporta al usuario final del producto y permiten responder rápidamente a los requisitos cambiantes.

Kanban:

Es un marco de trabajo ágil que requiere una comunicación en tiempo real sobre la capacidad y una transparencia del trabajo total. Se usa un tablero donde los elementos de trabajo se representan visualmente, lo que permite a los miembros del equipo ver el estado de cada uno y del proyecto o sprint en cualquier momento.

Arquitectura de referencia:

Una arquitectura de referencia está formada por un documento o un conjunto de documentos que ofrecen estructuras e integraciones recomendadas de productos y servicios de tecnologías de la información para formar una solución. La arquitectura de referencia incorpora las mejores prácticas aceptadas del sector, que normalmente sugieren el método de entrega o las tecnologías concretas óptimas.

Personas (UX):

Personas son personajes ficticios basados en datos reales del usuario, permiten describir los comportamientos, motivaciones, frustraciones y metas de los usuarios a los que está dirigido un producto o servicio. Estos ayudan a ilustrar comportamientos y tareas comunes que se comparten entre grupos de gente, ayudando a los diseñadores a imaginar cómo los usuarios podrían interactuar con un producto y qué características necesitan para lograr sus objetivos. Como herramienta UX, proporcionan una forma memorable de extraer y sintetizar la investigación de usuarios en comparación con una lista abstracta de requisitos de diseño, esta presentación más realista y centrada en los personajes también ayuda a fomentar la empatía, utilizando personajes ficticios para describir los hallazgos de una manera más humana y fácil de identificarse.

Benchmarking:

El benchmarking es el proceso de medir el desempeño de una empresa, aplicativo o cualquier componente en general al compararse con otro mejor o en diferentes condiciones. Es básicamente aprender de los demás utilizando el conocimiento y la experiencia de otros para mejorar o generar ideas nuevas.

PowerBI:

Es un servicio en de Microsoft donde se puede desarrollar rápidamente tableros, compartir informes y conectarse directamente a todos los datos importantes e incorporarlos. Éste muestra tableros interactivos que se pueden crear y actualizar desde muchas fuentes de datos diferentes, ya que además es intuitivo y más centrado en las necesidades del usuario, es capaz de analizar grandes

cantidades de datos y convertirlos en informes de gran riqueza visual útiles para detectar patrones o características que, de otro modo, sería muy costoso de conseguir.

OneDrive:

Es una plataforma creada por Microsoft que permite almacenar fotos, archivos y carpetas y compartirlos con otros usuarios. OneDrive también permite el acceso a los archivos al almacenarlos en la nube y sincronizarlos con distintos dispositivos. El espacio de almacenamiento ofrecido es hasta de 5 GB libres y el servicio incluye aplicaciones como Word, Excel, PowerPoint y OneNote.

Teams:

Microsoft Teams se basa en Grupos de Office 365 y permite la colaboración entre las personas de un mismo equipo o el desarrollo de un proyecto en particular, su función principal es la comunicación constante entre los miembros del equipo. Al crear un equipo en Microsoft Teams, se crea un tablero. Este tablero contiene una serie de funciones básicas, que son: chat, opciones de guardar y compartir archivos y la posibilidad de organizar reuniones dentro del propio equipo.

Jira:

Jira Software es una herramienta ágil de gestión de proyectos compatible con cualquier metodología ágil, ya sea scrum o kanban. Desde tableros ágiles hasta informes, Jira permite planificar y gestionar todos los proyectos de desarrollo de software ágil y hacer un seguimiento de ellos con una sola herramienta.

JasperReports:

Es una herramienta de generación de reportes desarrollado por Jaspersoft. Incluye tablas, gráficos, tableros, tablas cruzadas y medidores. Las fuentes de datos pueden provenir de archivos planos o XML y bases de datos relacionales. Es posible generar los reportes en una variedad de formatos, como PDF, HTML, HTML, RTF, XLS, CSV o XML. La Librería está escrita completamente en Java y puede usarse en una variedad de aplicaciones habilitadas para Java, incluyendo J2EE o aplicaciones web, para generar contenido dinámico. Su objetivo principal es ayudar a crear documentos listos para imprimir de una manera simple y flexible.

API (Application Programming Interface):

Conjunto de constantes, funciones y protocolos que permiten programar aplicaciones. Una buena API ayuda en la tarea de desarrollar pues permite que las funcionalidades de un servicio inicial puedan ser ampliadas a otras, pudiendo integrar otras APIs haciendo que el servicio/software sea extensible. En conclusión, las APIs facilitan considerablemente la construcción de aplicaciones más complejas, ya que facilita todas las piezas y el programador sólo tiene que unirlos para lograr el fin que desea.

Equipos en la dirección de vida y rentas:

Todos los miembros se encuentran divididos entre tres equipos, el equipo de proyectos que se encarga de los nuevos desarrollos, el equipo de mejora continua que se encarga de mantener y mejorar las aplicaciones existentes entre los negocios de vida y rentas, finalmente el equipo de operaciones que se encarga de los requerimientos cortos y la atención oportuna de incidentes generados por las aplicaciones de la misma dirección. Adicionalmente los equipos de mejora continua y operaciones se encargan en conjunto de resolver la raíz de las encontradas.

Raíz:

Con este término nos referimos a la causa raíz de un error en una aplicación, estas son generadas por el equipo de operaciones al resolver un incidente. Por lo tanto resolver una raíz implica que no se vuelvan a presentar errores por la misma causa.

Incidente:

Error reportado por el usuario final de la aplicación, el cual requiere de la ayuda del personal de tecnología para resolver el problema y así permitir que el usuario final continúe con sus labores normales.

Metodología

El primer paso fue definir lo que se buscaba comprender con el uso de los tableros de control, para ello se hicieron las siguientes actividades basadas en el ciclo iterativo de Design Thinking:

1. **Creación de Personas (UX).** Con esto se lograron definir las características principales de los clientes en el proyecto, para así enfocarse en ellos en las siguientes definiciones y durante todo el desarrollo del proyecto.
2. **Reuniones de definición.** En ellas se logró encontrar cuál información se debía mostrar en los tableros de forma que se obtuviera el mayor provecho para las Personas (UX).
3. **Benchmarking.** Dados los datos adquiridos en el punto anterior, se buscaron tableros que ya existieran para encontrar puntos comunes que posteriormente fueron usados en los prototipos o en el producto final entregado.
4. **Prototipado de baja fidelidad.** Se diseñaron componentes individuales para aclarar la forma de visualizar la información durante las reuniones de definición y así poder comprender mejor cómo sería el resultado final.

Luego del proceso inicial de definición se buscan qué tecnologías serían las más adecuadas para llevar a cabo el desarrollo del proyecto, pero teniendo como prioritarias las que ya sean usadas en la compañía, esto con el fin de disminuir en

costos de desarrollo y tiempo en implementación o integración de protocolos de seguridad.

Posteriormente empieza la fase de implementación usando Scrumban (Combinación entre el marco de trabajo Scrum y Kanban) la cual tuvo las siguientes partes:

1. **Generación de tareas de desarrollo.** Se crearon las historias de usuario de acuerdo a la información recolectada en la fase de definición.
2. **Obtención de información real.** Se buscó la forma de adquirir partes de la información real requerida para poderla usar en la herramienta seleccionada para el desarrollo y así lograr generar prototipos rápidos pero que fueran fácilmente validables.
3. **Prototipado de alta fidelidad.** Se crearon prototipos pequeños en la herramienta seleccionada para el desarrollo y con datos reales para intentar mostrar cómo sería el entregable final, de manera que se pudiera validar tanto la forma gráfica de entregar la información como la veracidad de los datos mostrados.
4. **Reuniones de validación.** Se realizaron varias reuniones de validaciones para determinar si las gráficas y los datos usados eran los adecuados para transmitir la información de manera acertada.
5. **Desarrollo en la herramienta seleccionada.** Aquí se empezó a desarrollar el producto final usando toda la información obtenida en los pasos anteriores, el proceso realizado se desglosa en estas tareas:
 - a. **Obtención de toda la información requerida.** Se identificaron e integraron las funciones requeridas para obtener la información necesaria en tiempo real para el producto final.
 - b. **Análisis de datos.** Se analizaron los datos para solo tomar lo necesario para la representación correcta de la información.
 - c. **Transformación de datos.** Se transformaron los datos obtenidos para que la herramienta de desarrollo pueda usarlos en la graficación de información.
 - d. **Optimización.** Se optimizaron los procesos anteriores para disminuir tiempos de carga y mejorar la interacción con el cliente.
 - e. **Creación de gráficos.** Se generaron los gráficos ya validados.
 - f. **Publicación del tablero de control.** Se publicó el tablero de control en las formas predispuestas por la compañía y que a su vez facilita la interacción con el cliente.

Resultados y análisis

Como parte del proceso de definición se crearon dos Personas (UX), uno que tuviera las características generalizadas de los miembros del equipo de trabajo y otro que tuviera las características de los líderes, pues en ambos se encontraron diferencias en cuanto a metas, objetivos y responsabilidades.

Para la Persona (UX) de los miembros del equipo se hizo una pequeña entrevista a 10 personas con el mismo cargo de Analista de Sistemas y la información encontrada quedó plasmada en la Persona 'Camilo' tal como se muestra en el gráfico a continuación (Figura 1).

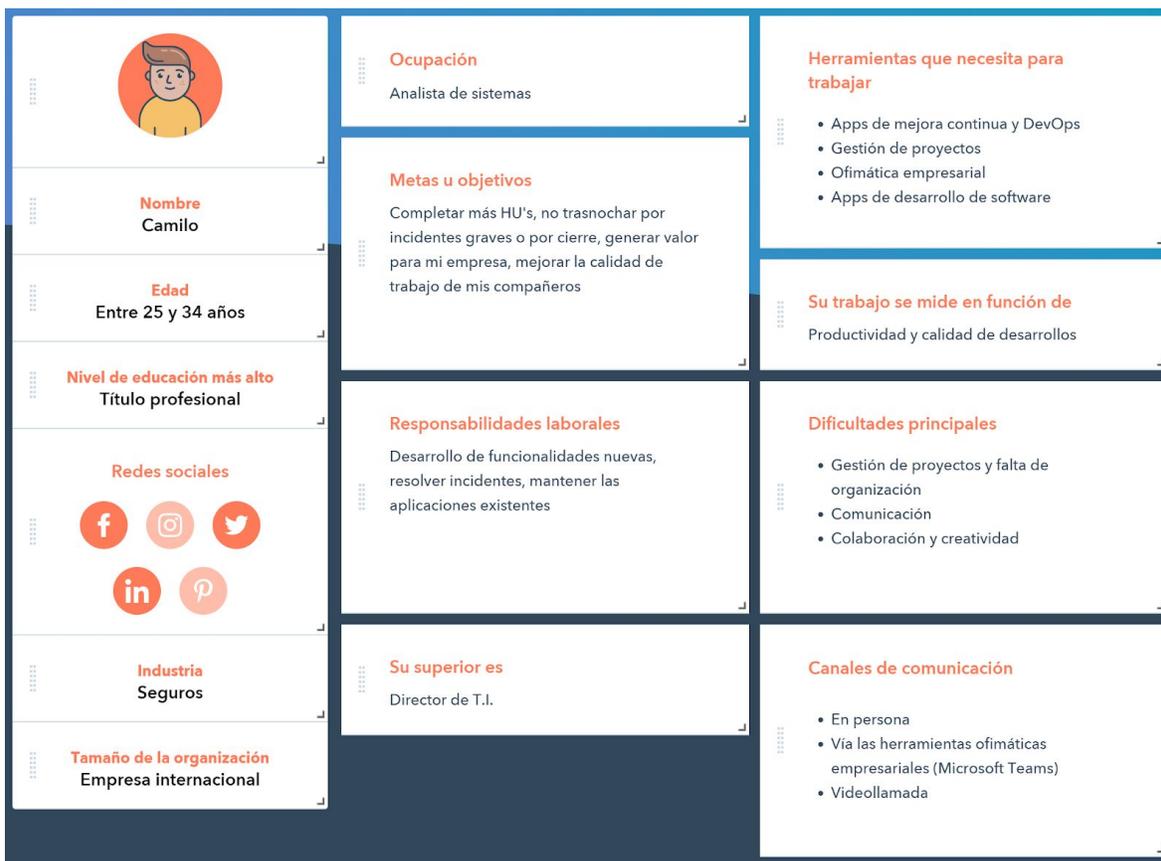


Figura 1. Persona (UX) representando al equipo de desarrollo

En esta Persona (UX) hubo un enfoque especial en la parte de mejorar la calidad de trabajo, pues una situación común que se presenta es que debido a errores en procesos masivos o por incidentes graves se debe trabajar un tiempo adicional para dar solución, en algunos casos incluso en la noche o madrugada. Esto lleva a que no se cumplan las metas propuestas para el sprint o que se acumule un cansancio innecesario en quien sea responsable de los procesos afectados.

En cambio para la Persona (UX) que fue creada con las características del líder de T.I y los Scrum Master, la diferencia principal con respecto a 'Camilo' es que en

sus objetivos está el generar accionables para mejorar los procesos de desarrollo del equipo, además de revisar constantemente cuál era el estado de los proyectos actuales para poder comunicarlo efectivamente a los miembros del negocio que en otras palabras son nuestros clientes.

Se debió considerar además que ambas Personas (UX) harían uso del mismo tablero, pero como sus objetivos son diferentes su forma de interpretar la información sería diferente. Un ejemplo es que para la persona representando al equipo de desarrollo ver la lista de raizales prioritarias es muy útil porque le permite generar un mayor impacto con la menor cantidad de desarrollos, mientras que para la persona representado a los líderes, le interesa más visualizar que cuando la cantidad de raizales vaya disminuyendo también disminuya de manera acorde la cantidad de incidentes cerrados para verificar que los procesos implementados sean los correctos.

Posteriormente en las reuniones de definición se habló específicamente de qué información sería valiosa para que estas Personas (UX) pudieran lograr sus objetivos. Algunos datos importantes a recolectar serían:

- El estado actual del sprint, de manera que se pudiera predecir aproximadamente si se cumplirían los objetivos en el tiempo restante.
- Los puntajes de las historias de usuario por cada proyecto para así tener conciencia del esfuerzo dedicado o la prioridad por negocio.
- El puntaje por historias de usuario terminadas en los últimos sprint para mejorar las planeaciones siguientes.
- La cantidad de incidentes mensuales, pues el histórico nos ayuda a ver si estamos mejorando constantemente o si en cambio estamos inyectando errores en nuestros desarrollos.
- La cantidad de incidentes abiertos actualmente, ya que puede darse el caso en que varios miembros del equipo de desarrollo deban apoyar al equipo de operaciones con los incidentes en caso que ellos tengan demasiada carga.

Dada esta información se hicieron prototipos rápidos en papel para validar la forma de representar los datos de manera que fuera más entendible para las Personas (UX), en esta actividad cualquier tipo de idea era válida y todos los implicados en tales reuniones tenían el mismo voto sin importar cual fuera su cargo real. Por ejemplo, para revisar el estado actual de las historias de usuario se propusieron tres tipos de gráficos (Figura 2) y además se definió si el conteo debería ser por cantidad de historias de usuario o por el puntaje de estas, finalmente se decidió por el gráfico circular y el puntaje por historia de usuario.



Figura 2. Prototipo en baja fidelidad del gráfico 'Estado del Sprint actual'

Al ya tener definida la información que se requería y los tipos de gráficos se comenzó a analizar la tecnología en la que se iba a crear el tablero de control, pero pensando siempre en cómo se le iba a entregar a nuestros clientes. Lo ideal sería que fuera en una aplicación web pues es de fácil acceso, sin embargo debido a los costos que acarrearía una aplicación web creada desde el inicio se optó por revisar lo que ya existía en la compañía para generar tableros de control, se encontraron los siguientes posibles candidatos:

1. **Jira:** Este permite generar tableros de control, sin embargo está algo limitado por el hecho de que solo se puede obtener información desde los ítems creados en Jira y además no tiene el manejo de datos personalizados necesario para los gráficos requeridos.
2. **JasperReports:** Se encontró que este permitiría generar todos los gráficos requeridos y también tiene la forma de transformar los datos desde distintas fuentes, sin embargo, en la compañía existen ya varios tableros creados en este software, dialogando con sus diseñadores se encontró que aunque la herramienta es poderosa es bastante compleja de utilizar y su interfaz gráfica no es tan moderna.
3. **PowerBI:** Se encontró que esta contiene una gran variedad de gráficas posibles y se permite integrar con múltiples fuentes de datos, sin embargo, lo más atractivo y el motivo por el cual se escogió esta herramienta sobre las otras, es que la compañía tiene una set ofimática de Microsoft y este software está incluido allí, por lo tanto todos los clientes tienen acceso a este aplicativo y se integraría con Microsoft Teams para la visualización del tablero de control además de tener su propia aplicación web para desplegar, visualizar y compartir tableros.

Las fuentes de datos necesarias que se debieron integrar con PowerBI fueron Jira y algunos libros de excel pues parte de la información necesaria se digitalizaba manualmente.

Para conectar Jira se hizo mediante su API, la cual devolvía un formato JSON con la información completa de cada ítem, por ejemplo, extraer la información de todas las historias de usuario de los últimos 5 sprints. Inicialmente se hicieron varios filtros en Jira para seleccionar sólo los ítems pertinentes, sin embargo se llegó a una limitante en la cual solo se podían traer un máximo de 1000 ítems por llamado a la API, entonces para obtener el historial de incidentes se tuvo que partir por meses para lograr obtener la información completa del año.

Con los libros de excel se aprovechó la integración que ya existía con el set de ofimática de Microsoft, entonces se pusieron los libros en carpetas compartidas generadas por OneDrive, las cuales se pueden acceder y editar en línea de manera que siempre se pudiera actualizar de manera sencilla y así mismo los cambios se vieran reflejados en el tablero de control en tiempo real. Desde PowerBI solo se necesitó acceder a estas rutas para traer la totalidad de la información de los libros.

Luego de conectar ambas fuentes de datos a PowerBI se tomó una pequeña porción de la información y se plasmó en un tablero generado por la misma herramienta para visualizar rápidamente su capacidad y poder validar con datos reales su efectividad. Un ejemplo de uno de los prototipos en alta fidelidad es el siguiente (Figura 3):

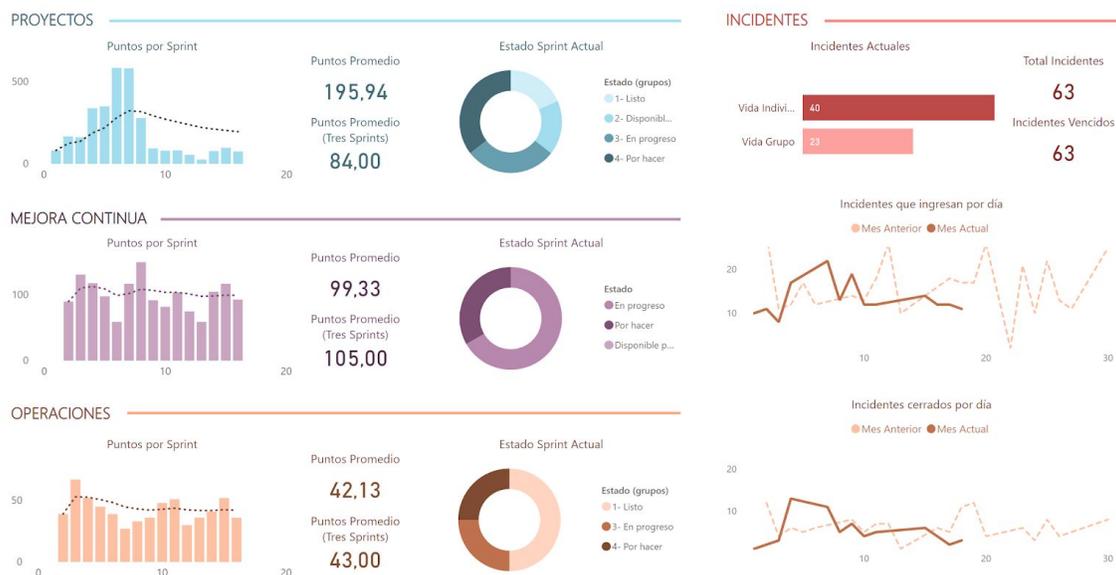


Figura 3. Prototipo en alta fidelidad de la versión inicial del tablero de control

De tal prototipo se obtuvieron conclusiones importantes como:

- No es necesario un histórico de todos los sprints existentes, pues los equipos cambian y evolucionan constantemente, entonces si los medimos en

comparación a hace 10 sprints los resultados no serían los adecuados para estimar lo que será del próximo sprint.

- Aunque tener la cantidad de incidentes que ingresan diariamente por mes puede ser útil para el equipo de operaciones, no es una información relevante para planear acciones de mejora, sería más útil ver como se ha comportado mes a mes.

Luego de varias iteraciones de prototipado y validación se llegó a un diseño final, entonces se pasó a optimizar los tiempos de carga de los datos pues se requería tomar más de 6.000 historias de usuario, que luego PowerBI transformaría para que los gráficos pudieran interpretar la información, por lo que cada actualización tardaba aproximadamente 12 minutos. Por este motivo se modificaron los conjuntos de datos para tomar lo estrictamente necesario, se cambió el orden de las transformaciones y se referenciaron entre sí otros conjuntos de datos. Con estos cambios el tiempo de actualización pasó a ser de 2 minutos aproximadamente.

Finalmente se llegó al último diseño del tablero, el cual tiene los siguientes ítems:

- Puntajes de historias de usuario finalizadas en los últimos 6 sprint por cada uno de los equipos de la dirección.
- Tendencia de completitud de historias de usuario de los últimos 6 sprints.
- Promedio de puntos por historias completadas en los últimos 3 sprints por cada uno de los equipos de la dirección.
- Estado de las historias de usuario del sprint en curso por cada uno de los equipos de la dirección.
- Puntaje de historias de usuario totales por proyecto (O por epica) y por cada equipo de la dirección, adicionalmente estas historias tienen su estado actual.
- Incidentes que fueron atendidos por el equipo de operaciones durante el año actual y el anterior.
- Incidentes que fueron resueltos (Cerrados) por el equipo de operaciones durante el año actual y el anterior.
- Incidentes que están pendientes por resolver (Abiertos) y deben ser gestionados por el equipo de operaciones, estos se clasifican por cada solución de la dirección (Vida Individual, Vida Grupo, Rentas).
- Listado de las raizales sin resolver, sin clasificación por solución y ordenadas desde la que ha generado más incidentes hasta la que menos ha generado.



Figura 4. Tablero de control unificado de la dirección vida y rentas

Luego de tener hecho el tablero se procedió a publicarlo y compartirlo, por lo cual se desplegó a un área de trabajo compartida en Teams por todos los miembros de la dirección, lo cual generó una nueva pestaña que daría acceso de lectura al tablero. Adicionalmente se programó desde la aplicación de PowerBI en la nube que éste se actualizara diariamente en horas de la noche para mantenerlo vigente con respecto a los datos recolectados y sin riesgo de afectar las aplicaciones transaccionales de las que toma la información.

Luego de la entrega del proyecto se pudo evidenciar en su uso por parte de todos los miembros de la dirección lo siguiente:

- Al tener la lista de las raizales más importantes la priorización para su resolución fue más sencilla.
- En las planeaciones se hacía una estimación manualmente con respecto a los puntajes de historias de usuario que se lograron desarrollar en los últimos sprints, con este tablero mejoró tal proceso e hizo más visibles los logros de cada equipo de desarrollo.
- Como todos los miembros de los equipos pueden conocer en tiempo real la cantidad de incidentes abiertos por solución, se ha logrado ayudar más efectivamente a solucionarlos en conjunto al contar con expertos en el tema específico que estaba siendo más importante.
- Como un equipo de desarrollo tiene que atender historias de usuario de varios proyectos, se pudo revisar que proyecto estaba usando más recursos que otros para poder balancear las cargas más equitativamente.

Conclusiones

- El uso de Personas (UX) ayuda a mantener un enfoque más preciso en los clientes durante cualquier etapa de desarrollo de un proyecto.
- Las entrevistas pueden brindar información importante para el desarrollo de un proyecto puesto que permite discutir las particularidades de las situaciones vividas que llevaron a la problemática que se quiere solucionar.
- El uso de prototipos en reuniones de definición permite precisar con mayor certeza lo que se espera con el desarrollo.
- Al usar prototipos de alta fidelidad hechos con el software con el que se espera desarrollar el proyecto se pueden encontrar falencias que de ser encontradas en etapas posteriores del desarrollo llevarían a mayores costos de implementación.
- Jira posee una API que permite integrar de manera sencilla cualquier aplicación que pueda interpretar y transformar datos que viajan en formato JSON.
- Al hacer uso de la herramienta de trabajo PowerBI que hace parte de una suite ofimática de Microsoft que integra tanto seguridad, como manejo de roles y permisos, además de archivos compartidos, se logra reducir considerablemente los costos de despliegue en comparación a una arquitectura en la nube hecha para el despliegue de una aplicación web. Sin embargo, esto se aplicó en este proyecto puesto que la compañía ya tenía tal suite ofimática contratada.
- Cuando todos los miembros de un amplio equipo de trabajo conocen sobre la generalidad de los procesos que se llevan a cabo en los demás equipos pueden surgir más ideas y accionables en pro del bienestar de los demás miembros.
- Al proporcionar la información clave con respecto al puntaje de historias listas de los últimos sprints terminados, se mejora el proceso de planeación de los nuevos sprints.
- Mantener una lista de los errores más críticos (Raizales) ayuda a priorizar la atención de su solución más efectivamente.
- Al tener visible constantemente el estado del sprint actual se permite predecir con más facilidad si los objetivos de tal sprint serán cumplidos.

Trabajo futuro

- Expandir los datos de historias de usuarios o puntajes por proyecto a presupuesto invertido por proyecto.
- Proporcionar tableros adicionales para uso exclusivo del negocio y con sólo información pertinente para ellos.
- Adicionar gráficos de análisis de entradas a bases de datos productivas para corrección de incidentes.

Referencias

1. Acamica, Carrera de Diseño UX/UI. [En línea]. Disponible en: www.acamica.com
2. H. Plattner, C. Meinel, L. Leifer (2010) Design Thinking: Understand – Improve – Apply
3. Usability.gov, Prototyping. [En línea]. Disponible en: <https://www.usability.gov/how-to-and-tools/methods/prototyping.html>
4. Platzi, Qué es el Frontend y Backend. [En línea]. Disponible en: <https://platzi.com/blog/que-es-frontend-y-backend/>
5. Atlassian, Atlassian Agile Couch. [En línea]. Disponible en: <https://es.atlassian.com/agile/>
6. Hewlett Packard, Hewlett Packard Enterprise. [En línea]. Disponible en: <https://www.hpe.com/es/es/what-is/reference-architecture.html>
7. Volentine, R., Whitson, L., & Tenopir, C. (2012). "Portraits of Success: Building Personas from Scholarly Reading Patterns". QQML 4th annual conference proceedings, 2017. [En línea]. Disponible en: <http://qqml-journal.net/index.php/qqml/article/view/72/75>
8. A. Sundt and E. Davis, "User Personas as a Shared Lens for Library UX", Weave Journal of Library User Experience, vol. 1, no. 6, 2017. [En línea]. Disponible en: <https://doi.org/10.3998/weave.12535642.0001.601>
9. W. Lankford. "Benchmarking: Understanding the basics." The Coastal Business Journal 1.1. 2000, pp. 57-62.
10. V. NEGRUT, "POWER BI: EFFECTIVE DATA AGGREGATION", Quaestus, vol. 13, pp. 146--152, 2018.
11. B. Meseguer Barrionuevo, "El business intelligence en las PYMES : herramienta power BI", Repositorio.upct.es, 2016. [En línea]. Disponible en: <http://repositorio.upct.es/handle/10317/5728>
12. A. Kuz, M. Falco, y R. Giandini, Comprendiendo la Aplicabilidad de Scrum en el Aula: Herramientas y Ejemplos, TEyET, n.º 21, p. e07, jun. 2018.
13. C. Julieta, S. Esponda, and M. Boracchia, "Hacia una mejora de calidad en Scrum: integrando documentación requerida por IRAM-ISO 9001: 2015." XXIV Congreso Argentino de Ciencias de la Computación. La Plata, 2018
14. K. Villamizar Suaza, J. Tabares García, y C. Zapata Jaramillo, "Mejora de historias de usuario y casos de prueba de metodologías ágiles con base en TDD"., CA, vol. 7, n.º 1, pp. 41-53, 1.
15. Ayuda de OneDrive: soporte técnico de Office", Support.office.com, 2020. [En línea]. Disponible en: <https://support.office.com/es-es/onedrive>
16. O. Huarte, "¿Qué es Microsoft Teams?", Soluciones-sharepoint.com. [En línea]. Disponible en: <https://www.soluciones-sharepoint.com/2017/09/microsoft-teams.html>
17. S. Luján Mora, Programación de aplicaciones web. San Vicente (Alicante): Editorial Club Universitario, 2002.
18. G. Soto, Nuria. "Desarrollo De Una API Para Datos Abiertos". Repositorio institucional de la Universidad de La Laguna, 2018, [En línea] Disponible en: <https://riull.ull.es/xmlui/handle/915/7106>.
19. Jira, Software de seguimiento de proyectos e incidencias", Atlassian, 2020. [En línea]. Disponible en: <https://www.atlassian.com/es/software/jira>.

20. T. Danciu, "The JasperReports Ultimate Guide," ed, 2002.
21. "JasperReports® Library", Jaspersoft Community, 2020. [En línea].
Disponible en:
<https://community.jaspersoft.com/project/jasperreports-library>.