***Proyecto practica AllCode***

***Por:***

*Isaías Labrador Sánchez*

***Proyecto realizado en***

*AllCode*

***Presentado a***

*EAFIT*

*Medellín - 2022*

***Introducción***

En las empresas de software generalmente se busca mantener una relación cercana con el cliente, que conozca sus productos, se involucre en el desarrollo y sepa en todo momento cuales son las prioridades para el equipo de desarrollo. En AllCode, parte de esta relación con el cliente incluye reportar las actividades que los recursos desarrollaron durante cierto rango de tiempo, para así el cliente saber en que invierte su dinero y los lideres de grupo controlan el avance del proyecto.

Teniendo en cuenta el proceso manual por el que tienen que pasar los desarrolladores cada que tienen que reportar actividades, se piensa en automatizar ciertos procesos para hacer todo esto mucho mas sencillo y ameno para los desarrolladores, lideres de equipo y clientes. Se piensa en una herramienta web con portabilidad móvil que permita el manejo de actividades dentro del ambiente de desarrollo de la empresa, facilitando el manejo de proyectos, recursos y tareas.

***Justificación***

Esta solución de software enfocada en la automatización del proceso de reporte y manejo de las tareas dentro de un proyecto nace con la motivación de brindar a los desarrolladores y lideres de equipo una herramienta que facilite el ingreso de horas trabajadas asociadas a una tarea y un proyecto específico. Esta herramienta brindara la posibilidad de asignar recursos y prioridades dentro de los equipos de desarrollo, teniendo el plus además de ser una herramienta de uso interno, lo que hace que el mantenimiento y la adaptación de procesos de la solución sea mucho más flexible y valiosa para toda la empresa, manteniendo similitud a los procesos que ya se venían haciendo en los proyectos desde antes con los desarrolladores.

También se busca mantener o dejar abierta la posibilidad de relacionar las actividades o tareas del proyecto con las herramientas que los clientes prefieran para manejar problemas y tareas de su producto, tipo JIRA, ClickUp, Basecamp, etc. Lo que quiere decir que se pueden relacionar tareas y actividades de dichas plataformas con los reportes de la solución aquí presentada.

***[Explicar el origen]***

***Objetivos***

Se plantean aquí los objetivos diferenciando los que se quieren alcanzar con la culminación o a través del desarrollo y mantenimiento de la solución y aquellos objetivos que se quieren alcanzar durante el periodo de validación de la práctica, teniendo en cuenta que el tiempo de desarrollo, planeación e investigación (±3 meses) no es suficiente para el desarrollo completo de la solución y sus requisitos funcionales y no funcionales, así como la aplicación de la investigación de diseño orientado a la experiencia de usuario.

***Objetivos:***

* Desarrollar una herramienta que automatice el reporte de actividades y tareas de los desarrolladores dentro de la empresa.
* Desarrollar una herramienta que permita la gestión de desarrolladores por parte de los líderes de proyecto.
* Sincronizar reporte de actividades desde la solución presentada hasta la plataforma de freshbooks.

***Objetivos específicos:***

* Realizar la definición del proyecto y capturar los requisitos.
* Definir la arquitectura de la solución de reporte de actividades.
* Realizar la investigación del diseño orientado a la experiencia de usuario para capturar requisitos fundamentales para el usuario dentro del sistema.
* Realizar e implementar técnicas de modelado para la documentación, desarrollo y optimización de la solución.
* Tomar decisiones de herramientas y diseño de la aplicación.
* Implementar reporte de horas/actividades dentro de un ambiente de desarrollo con las funcionalidades básicas para identificar al usuario y reportar y manejar actividades y proyectos.

En este caso los alcanzables durante el proceso de validación de practica serían los objetivos específicos. Esto teniendo en cuenta que el desarrollo de la aplicación se espera continuar después de finalizado este periodo, gracias al valor que aporta y la importancia real de la implementación de una solución de este tipo.

***Marco Teórico y de referencia***

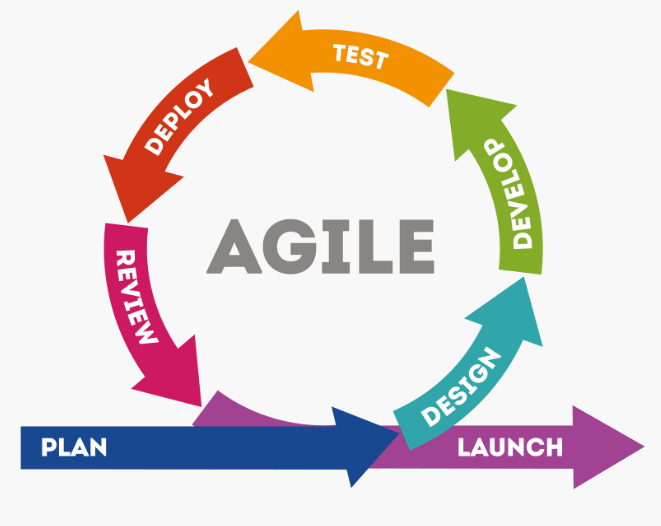
***Conceptos técnicos***

Aquí se aclararán de manera breve algunos conceptos técnicos utilizados para la definición e implementación de la solución presentada para la automatización del reporte de horas.

***Metodología Ágil***

Es un conjunto de principios o técnicas de desarrollo de proyectos, específicamente de software, que permite orientar todo el proceso de un proyecto a presentar soluciones al cliente que generen valor y que sean flexibles al cambio. Este concepto de desarrollo ágil surge a partir del ‘Manifiesto Ágil’ **[1]** que nos propone 12 principio para orientar el desarrollo a los resultados, recibiendo retroalimentación constante del cliente, validación de los requisitos y cambios rápidos en las reglas de negocio, que en otros contextos de desarrollo podrían dificultar las entregas y el aporte de valor. Aunque no todo es bueno, pues la metodología ágil supone un equipo ágil, disponible, proactivo y con una comunicación constante entre si y con el cliente **[2]**.

La siguiente figura representa el proceso de desarrollo de software general de la metodología ágil:

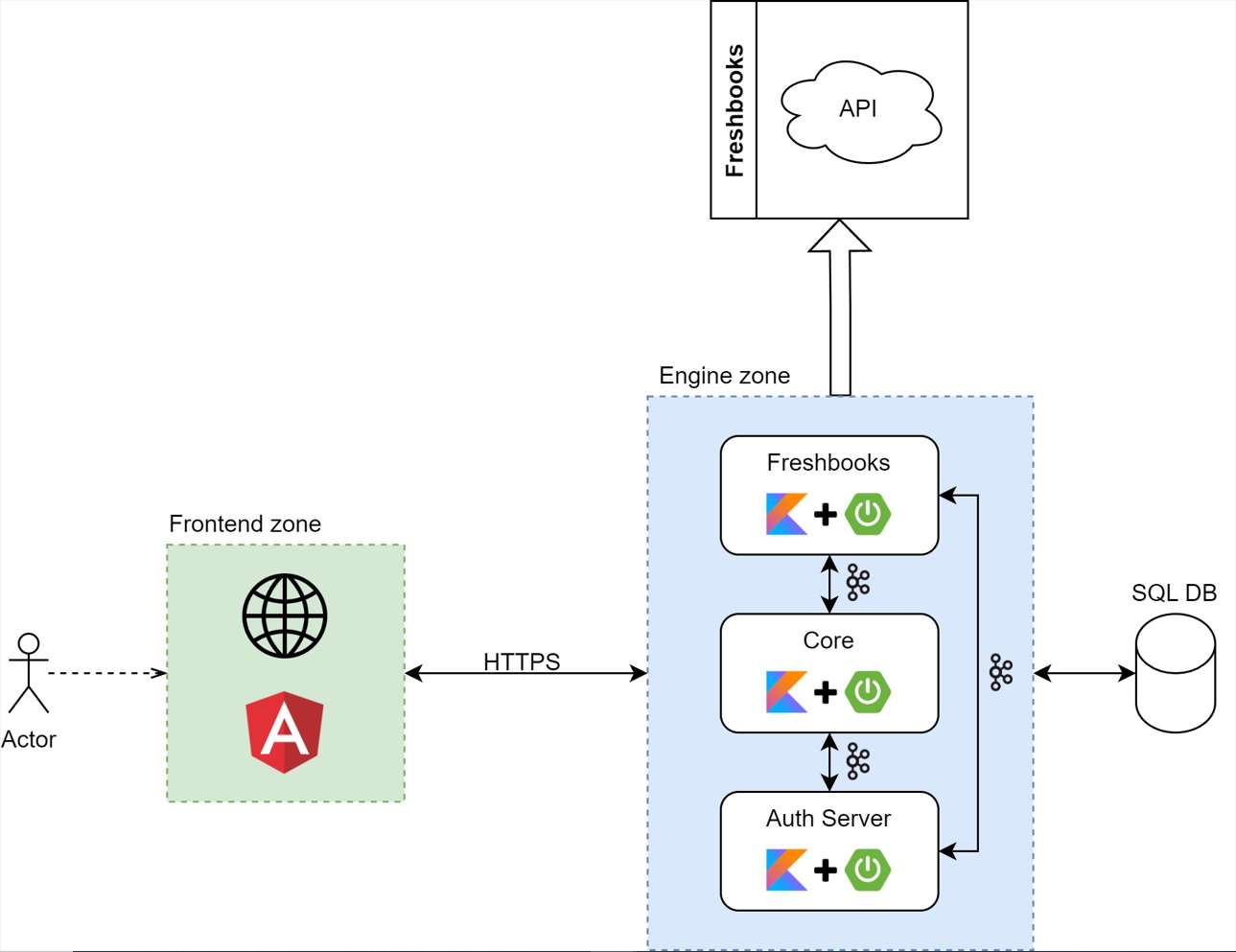


***Figura 1 [6]. 12 principios de la metodología agiles en el desarrollo de proyectos***. (s. f.). Cognodata. <https://www.cognodata.com/principios-metodologia-agile-desarrollo-proyectos/>

***Arquitectura de servicios***

La arquitectura orientada a servicios es un tipo de diseño de software que permite reutilizar sus elementos gracias a las interfaces de servicios que se comunican a través de una red con un lenguaje común **[3]**. En este caso tendremos a disposición 3 servicios comunicándose y colaborando entre sí: el de autenticación, el motor de las reglas de negocio y el modulo que sincroniza el reporte local del sistema con el aplicativo de freshbooks.

La arquitectura orientada a servicios permite la flexibilidad y usabilidad que se busca del proyecto permitiendo la independencia de las características de cada modulo y garantizando la comunicación entre las mismas. También este modelo permite la reducción de costos, acortando el tiempo de desarrollo, garantizando mayor eficiencia en el desarrollo y una mayor agilidad al cambio.



**Figura 2**. Arquitectura conceptual de alto nivel de la solución propuesta.

***Modelo MVC***

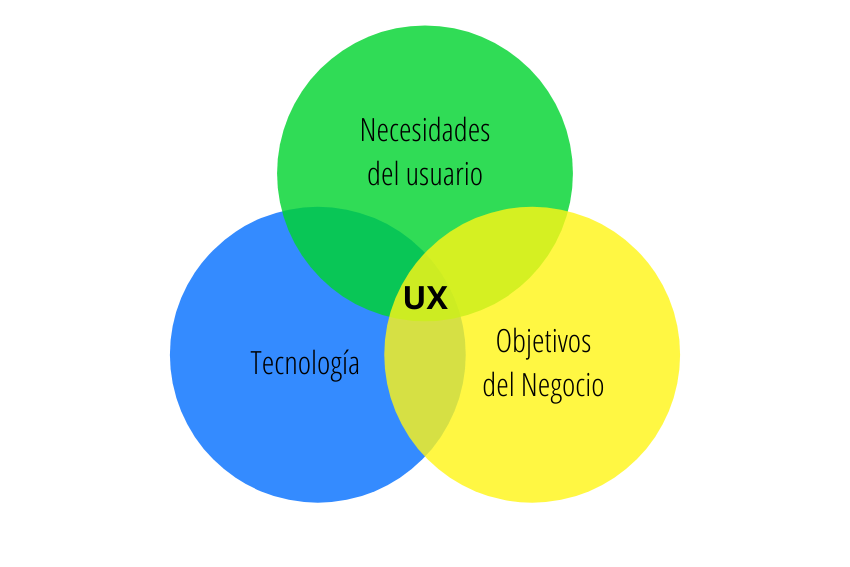
También en la arquitectura de la solución se implementa un concepto que ayuda a mantener la separación de procesos y garantiza la persistencia de los datos en un ambiente protegido: que es la arquitectura MVC (Modelo, Vista, Controlador). Aunque esta se define como una arquitectura aparte, ciertos aspectos de esta se implementan de manera implícita en el proyecto. En este caso el principio fundamental del modelo MVC es que separa en tres capas la solución **[4]**.

* **Modelo**: Capa encargada de la persistencia y gestión de datos (Bases de datos)
* **Vista**: Capa que contiene la interfaz gráfica. Generalmente se refiere a la interfaz con la que el usuario final interactúa.
* **Controlador**: Pate lógica del sistema. Encargado de implementar las reglas de negocio del sistema.

Este principio de las capas separadas se implementa en la solución para garantizar un ambiente seguro y reusable.

***Diseño de la experiencia de usuario***

Es un proceso que captura las actividades que realiza el usuario para llevarlas a una solución de software que sea lo más amigable e intuitiva para el usuario final. Todo el proceso da cuenta de lo que es relevante y natural para el usuario final permitiendo de esta manera tomar decisiones de diseño que impacten la solución de manera positiva y que genere en el usuario una sensación de valor de la solución aportada.



**Figura 3 [5].** Lucca, R. (s. f.). Diseño UX: Cómo mejorar la experiencia de usuario de tu web. DXMedia. <https://dxmedia.net/ux-mejorar-experiencia-usuario/>

El diseño orientado a la experiencia de usuario comprende varias etapas, y dependiendo del diseñador y la empresa se toman distintos caminos, pero en resumen se necesita recolectar información de:

* **Investigación del usuario**: En esta etapa se busca sacar información sobre el usuario. Toda la que sea relevante, a través de entrevistas, encuestas, contextual inquiry, entre otras. Todo esto con el fin de definir los intereses del usuario, la forma de llevar a cabo actividades, sus motivaciones, sus preocupaciones, quien es realmente el usuario. Ya que esto ayuda a definir una solución de mas valor para el usuario final.
* **Definición puntos críticos:** En esta etapa del proceso los diseñadores tienen la tarea de definir, a partir de la investigación del usuario, cuáles serán los pilares que la solución debe contener para así garantizar un enfoque correcto en el desarrollo e implementación de la solución.
* **Esquematización y navegabilidad:** Después de definir el usuario, sus intereses y aparte tener claros los puntos críticos del público objetivo, se procede a una esquematización. Que es donde se usa el resultado de las anteriores etapas para definir los pasos de las actividades, la navegabilidad y el diseño de la solución final. Se hace teniendo en cuenta las anteriores etapas ya que esto permite tener al usuario como pilar del desarrollo y diseño.
* **Diseño y prototipado:** En esta etapa se puede proceder a diseñar un prototipado a partir de la investigación que se le realizo al usuario. Al ser una etapa aún muy temprana y que no consume casi tiempo, se pueden tomar decisiones que hagan volver a cierta parte del proceso a ajustar algún error.

***Conceptos técnicos***

Aquí se explicarán los conceptos relacionados a la solución especifica de reporte de horas y actividades para los desarrolladores de AllCode.

***Entidades***

Las entidades básicas para comprender el proyecto son:

**Proyecto**: Esta entidad será la encargada de identificar dentro del sistema los diferentes proyectos de software. Sera la entidad encargada de identificar los permisos de cada usuario frente a si mismo.

**Tarea**: Las tareas aparecen relacionadas a los proyectos y son asignadas a los usuarios. Estas tareas pueden ser identificadas con un tipo, un responsable y una prioridad dentro del proyecto.

**Usuario**: Serán los que accedan al sistema y podrán realizar acciones sobre el mismo. Para el sistema actualmente solo se contemplan dos roles:

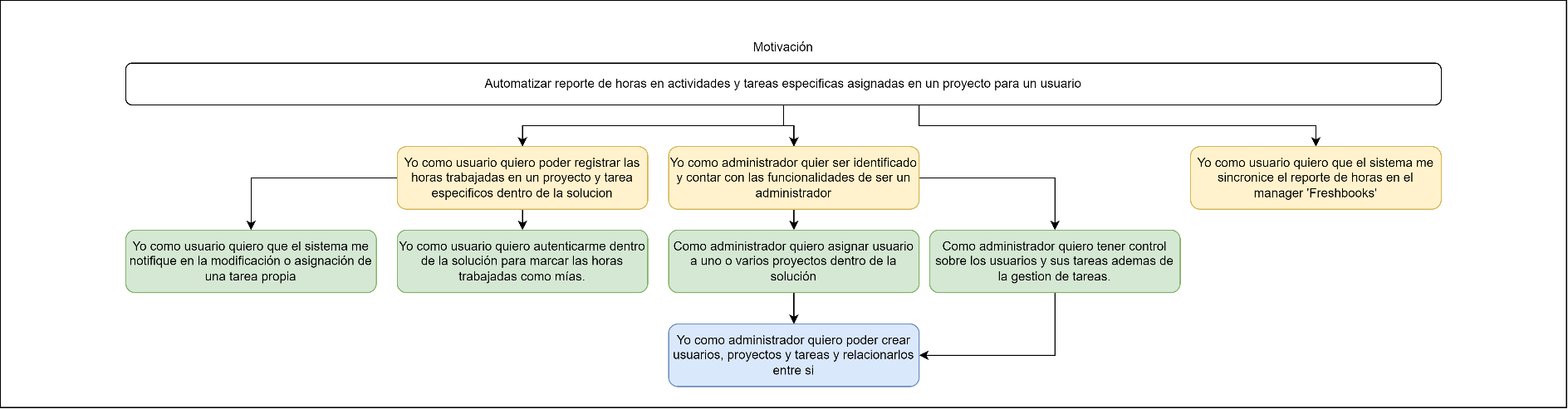
* **User:** Es el usuario con los permisos y funcionalidades básicas de la aplicación. A este se le asignara a un proyecto y distintas tareas.
* **Admin:** El usuario con rol de administrador cuenta con funcionalidades mas privilegiadas de gestión de usuarios, proyectos y tareas.

***Desarrollo***

Aplicando la metodología ágil y el diseño orientado a la experiencia de usuario se busca captar las funcionalidades importantes del sistema, su arquitectura, herramientas y además se toman decisiones de diseño de interfaz y navegabilidad. Todo esto con la validación continua con el usuario y la empresa (AllCode) permite avanzar en el proceso de desarrollo y diseño de la solución.



Durante los meses de septiembre a noviembre (2022) se busca generar la esquematización y modelación básica de la solución para llegar a un PMV (Producto Mínimo Viable) que satisfaga la condición de manejar el reporte de horas y la autenticación y gestión de usuario.



Con ayuda del jefe técnico de la empresa se captan aspectos importantes de la arquitectura y las herramientas a utilizar y de mi parte empiezo de manera autónoma la investigación del usuario final, que en este caso vendrían siendo los desarrolladores y los lideres de proyecto de la empresa.

Con la parte inicial del diseño y la arquitectura de la solución se empieza por prototipar las funcionalidades básicas por medio de REST API logrando un manejo de roles de usuario. Y con la parte de diseño de usuario y navegabilidad se procede a realizar la interfaz gráfica de usuario.

***Presentación y análisis de resultados***

Se logro implementar la definición de roles dentro del sistema además de un dashboard inicial que ya tiene definida la navegación como debería ser según la investigación del usuario, permitiendo implementar la esquematización sugerida y la autenticación del usuario. El reporte de horas en la base de datos del sistema aun se encuentra en desarrollo en este momento, pero el sistema ya permite la identificación y relación de usuarios, tareas y proyectos.

Ver Repositorios de la solución

* <https://bitbucket.org/jxmike/hh-frontend/src/master/>
* <https://bitbucket.org/jxmike/hh-core/src/master/>
* <https://bitbucket.org/jxmike/hh-auth/src/master/>

ESTOS REPOSITORIOS NO SON PUBLICOS (DE LA EMPRESA)

EJEMPLO DE LAS FUNCIONALIDADES IMPLEMENTADAS Y EL ANALISIS DE RESULTADOS DURANTE LA REUNION CON EL JEFE Y EL ASESOR DE PRACTICA

***Bibliografía***

1. Principios del Manifiesto Ágil. (s. f.). Recuperado 31 de octubre de 2022, de <https://agilemanifesto.org/iso/es/principles.html>
2. Silva, D. da. (2022, 19 febrero). ¿Qué es la metodología ágil? ¿Para qué sirve? Zendesk MX. <https://www.zendesk.com.mx/blog/metodologia-agil-que-es/>
3. ¿Qué es la arquitectura orientada a los servicios? (s. f.). Redhat. Recuperado 31 de octubre de 2022, de <https://www.redhat.com/es/topics/cloud-native-apps/what-is-service-oriented-architecture>
4. Modelo vista controlador (MVC). Servicio de Informática ASP.NET MVC 3 Framework. Universidad de Alicante. (s. f.). Recuperado 1 de noviembre de 2022, de <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
5. Lucca, R. (s. f.). Diseño UX: Cómo mejorar la experiencia de usuario de tu web. DXMedia. <https://dxmedia.net/ux-mejorar-experiencia-usuario/>
6. 12 principios de la metodología agiles en el desarrollo de proyectos. (s. f.). Cognodata. <https://www.cognodata.com/principios-metodologia-agile-desarrollo-proyectos/>