**Sistema de Trackeo de Tickets en tiempo real del Comedor de UNMSM (GT)**

**Reporte del Primer Sprint**

**Lima, 22 abril del 2023**

**Índice**

**Contenido**

[**1. Resumen 3**](#_nnlcpluz4xjv)

[**2. Abstract 3**](#_ot7nt9nd3csh)

[**3. Planteamiento del problema 4**](#_joy2hpcthhf9)

[**4. Justificación 4**](#_c746q4svsii0)

[**5. Objetivos 4**](#_nlz92fj522z4)

[5.1. Objetivo General 4](#_ls9ll93szhrb)

[5.2. Objetivos específicos 5](#_bgvqlalfxmd3)

[**6. Marco Conceptual 5**](#_6y5iu36nljow)

[6.1. Metodología Ágil Scrum 5](#_kbzp3k40r7ap)

[6.2. Metodología Cascada 6](#_lj7s0da7y30f)

[**7. Metodología 7**](#_flh1wmuwyc)

[**8. Ejecución 8**](#_ws8msh3bqc1x)

[8.1. Definición de Roles 8](#_xqyyf4gczzuf)

[8.2. Definición de Requerimientos 11](#_72jl1aoxwtaw)

[8.3. Diseño 12](#_w73nc3fsn62y)

[8.4. Desarrollo por Sprints 13](#_m9lokzkf276p)

[8.4.1. Sprint 1 14](#_ki0vfugvfru2)

[**9. Conclusiones 16**](#_9daxcgzae0jt)

[**10. Referencias 17**](#_hgw8j53y1tpi)

# 

# **Resumen**

En este trabajo se detalla tanto la planificación como el desarrollo de una plataforma web que tendrá el sistema de Trackeo de Tickets en tiempo real del Comedor de UNMSM, así también como el impacto que ha tenido la implementación de la metodología ágil Scrum en la ejecución de proyecto de software. Desde la perspectiva de la metodología, se ha analizado el progreso en la organización y productividad de cada uno de los miembros del equipo involucrado en el proyecto.

La plataforma tiene como objetivo poder ayudar con la organización de la emisión de tickets en el comedor de la UNMSM, sin la intervención de intermediarios o terceros, de este modo se puede lograr una gestión de la información de los platillos ofrecidos por día y la cantidad de alumnos que utilizan el servicio.

# **Abstract**

This paper details both the planning and the development of a web platform that will have the Ticket Tracking system in real time of the UNMSM Dining Room, as well as the impact that the implementation of the agile Scrum methodology has had on the execution of software project. From the perspective of the methodology, the progress in the organization and the productivity of each one of the members of the team involved in the project have been analyzed.

The platform aims to be able to help with the organization of the issuance of tickets in the UNMSM dining room, without the intervention of intermediaries or third parties, in this way, information management of the dishes offered per day and the quantity can be achieved. of students using the service.

# **Planteamiento del problema**

La falta de un sistema de trackeo de tickets en tiempo real puede causar varios inconvenientes en la atención a los estudiantes de la UNMSM y en la gestión del comedor. Uno de los principales problemas es la falta de comunicación entre los encargados de entrega de tickets y los alumnos, ya que esta falta de coordinación genera grandes filas y errores en la entrega de tickets, pues puede haber personas que realizan largas colas desde muy tempranas horas y aún así quedar sin un turno para la entrega de comida.

Además, la falta de un sistema de seguimiento en tiempo real de los tickets de los usuarios también puede generar disgustos en los alumnos, quienes no lograron obtener un puesto para el ticket, lo que puede resultar frustrante, se identifican dificultades para estimar plazo de espera de los alumnos lo que terminaría por generar pérdida de valioso tiempo.

Ante el problema expuesto Macro System, empresa que brinda soluciones personalizadas con respecto a temas de Software, desarrolla el proyecto basado en las necesidades expuestas por los usuario, es por ello que trabaja el sistema con debida planificación, cumpliendo los requerimientos estipulados, con los métodos apropiados que garanticen el orden durante toda la ejecución del proyecto mencionado.

# **Justificación**

La implementación del sistema planteado presenta múltiples justificaciones importantes, las cuales son:

1. Eficiencia en la gestión del comedor: La aplicación del sistema de trackeo de tickets permitirá un mejor control y seguimiento de la cantidad de alumnos en relación a la disponibilidad de turnos de una forma más eficiente
2. Mayor satisfacción de los alumnos o usuarios: Al contar con la información en tiempo real sobre el estado de la cantidad de turnos de comida y tiempo de espera, los usuarios tendrán una experiencia más agradable en el comedor y se sentirán más satisfechos con el servicio brindado.
3. Reducción de tiempo de espera: El sistema permitirá a los usuarios ver en tiempo real la cantidad de personas que están siendo atendidos en el comedor de la universidad, por lo que les dará la facultad de tomar decisiones informadas sobre cuándo acercarse o si alcanza un turno y de este modo reducir tiempos de espera o evitar hacer colas innecesarias.

# **Objetivos**

## **Objetivo General**

Es lograr implementar un sistema que permita el seguimiento del proceso de atención de cada uno de los pedidos de los comensales y facilite la gestión del personal encargado del comedor. El sistema deberá garantizar la eficiencia en la entrega de alimentos, así como también el control y la transparencia en el manejo de la información relacionada con el servicio del comedor.

## **Objetivos específicos**

1. Identificar los requisitos y necesidades del comedor de la UNMSM en cuanto a subsistema de atención a los alumnos y gestión de la información
2. Diseñar un interfaz intuitiva y amigable para el usuario final, que permita el acceso al número total de tickets de manera sencilla y la visualización de la información en tiempo real.
3. Desarrollar un sistema de seguimiento de tickets que permita al personal del comedor conocer el estado de la cantidad de tickets emitidos en tiempo real y garantizar la eficiencia en la entrega de los alimentos.
4. Capacitar al personal encargado del comedor en el uso adecuado del sistema de trackeo de tickets y asegurar su correcto funcionamiento.
5. Evaluar y ajustar el sistema de manera constante, a fin de garantizar su eficacia y eficiencia en la gestión del servicio de comedor de la UNMSM.

# **Marco Conceptual**

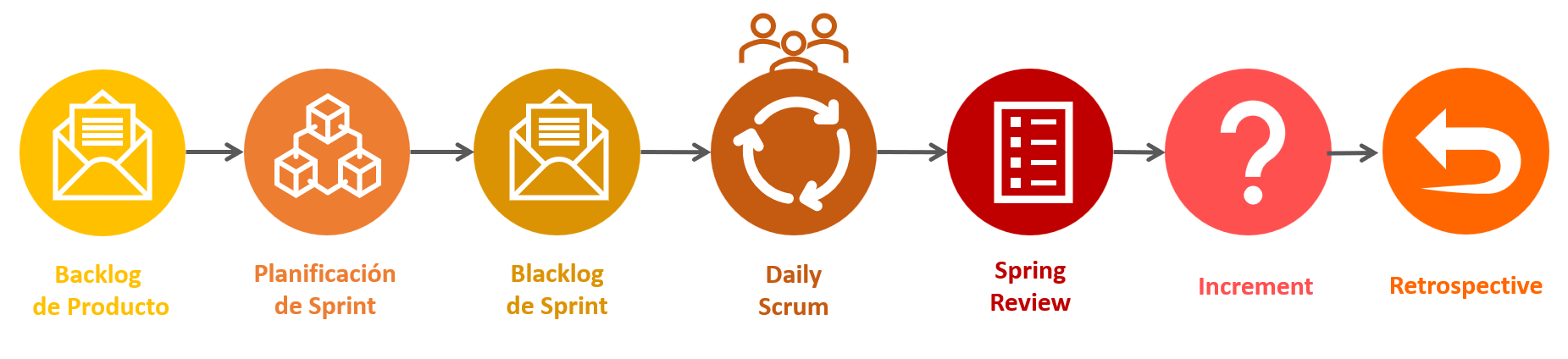
## **Metodología Ágil Scrum**

Scrum se enfoca en mantener una buena organización durante la construcción del producto y se basa en las historias de los usuarios para especificar cada requisito. Las denominadas Historias de Usuario forman parte de la primera etapa de Scrum, el *Product Backlog*, que es responsabilidad del Product Owner. Una vez disponible el *product backlog*, se establece el sprint backlog mediante métodos de estimación que permite establecer las tareas a realizar.

Una de las ventajas es el seguimiento constante del proyecto con reuniones diarias o *Daily 's*, en las que el equipo de desarrollo expone lo que se ha hecho, lo que se hará y las dificultades que se presentaron. De acuerdo con la realización de la supervisión continua, se utilizan grafos Burn-Down y Burn-Up, los cuales reflejan la gestión y avance del proyecto en cada sprint. Al final de estos se realiza una retrospectiva para evaluar el desempeño del equipo y corregir errores para el siguiente sprint.

**Figura 1.**

Metodología Scrum.



*Nota.* Elaboración propia.

Trabajar bajo la metodología Scrum aporta mayor flexibilidad y eficiencia a los proyectos que surgen en entornos versátiles y cambiantes, permitiendo una adaptación al cambio más rápida y eficaz. Además, la figura del Scrum Master es necesaria para asegurar el cumplimiento de las tareas y el buen funcionamiento del equipo.

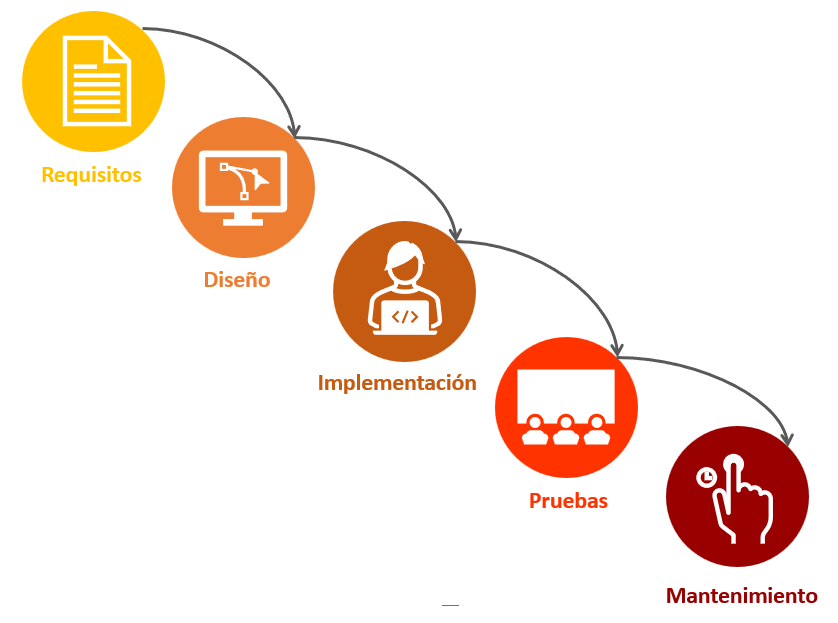
## **Metodología Cascada**

La metodología en cascada, también conocida como modelo de ciclo de vida de software en cascada, es un enfoque lineal y secuencial para el desarrollo de software. En este modelo, el proceso de desarrollo de software se divide en fases discretas y consecutivas, que deben completarse en orden antes de avanzar a la siguiente fase. Las fases típicas del modelo en cascada son:

* **Requisitos:** Recopilar y documentar los requisitos del software.
* **Diseño:** Crear y diseñar una arquitectura detallada del sistema.
* **Implementación:** Codificación del software.
* **Pruebas:** Verificar que el software cumple con los requisitos especificados y se detectan y corrigen errores.
* **Mantenimiento:** Realizar actualizaciones y correcciones al software ya implementado.

**Figura 2.**

Metodología Cascada.



*Nota.* Elaboración propia.

Una vez que se completa una fase, se pasa a la siguiente fase en secuencia. La metodología en cascada se basa en la idea de que el proceso de desarrollo de software es predecible y que se pueden especificar todos los requisitos antes de comenzar la implementación. Sin embargo, este enfoque puede ser inflexible y no tener en cuenta cambios en los requisitos o problemas imprevistos durante el desarrollo.

# **Metodología**

La metodología utilizada para el proyecto es una combinación de la metodología ágil Scrum y la metodología tradicional de Cascada. Esta combinación permite tener un enfoque más flexible y adaptable, mientras se mantiene la estructura y planificación necesarias para lograr los objetivos del proyecto.

La metodología ágil Scrum se enfoca en desarrollar productos de forma iterativa e incremental, en ciclos llamados Sprints. En específico, cada Sprint va a contar con una duración aproximada de 25 días y al final de cada uno se entrega un incremento del producto representado por la línea base. Durante el Sprint se llevan a cabo diferentes actividades como la planificación, el desarrollo, la revisión y la retrospectiva.

Por otro lado, la metodología tradicional de Cascada se enfoca en desarrollar productos en un proceso secuencial, con diferentes fases que se completan antes de avanzar a la siguiente. De esta forma, cada fase debe completarse antes de avanzar a la siguiente, y los cambios en una fase pueden tener un gran impacto en el desarrollo de las siguientes.

En la metodología utilizada en el proyecto, se sigue principalmente el flujo de Cascada, pero se incorporan elementos de Scrum. Se establecen tres líneas base o hitos importantes del proyecto y se llevan a cabo Sprints antes de cada uno de ellos. Al final de cada Sprint se lleva a cabo una retrospective con el objetivo de identificar oportunidades de mejora y hacer ajustes para el próximo Sprint.

* SPRINT RETROSPECTIVE: Se realiza una actividad retrospectiva para evaluar al equipo de desarrollo e identificar las áreas en las que se ha hecho bien, las que no se deben seguir haciendo y las que se pueden mejorar.

# **Ejecución**

## **Definición de Roles**

Para el desarrollo del proyecto: Sistema de Trackeo de Tickets en tiempo real del Comedor de UNMSM (GT) se establecieron linealidades para tomar decisiones respecto al enfoque que se debe de tomar.

* Las líneas base, que representan el fin de los sprints, tienen una duración aproximada de 25 días cada uno. Dentro del cronograma se definen actividades de análisis, desarrollo, implementación y pruebas.
* El equipo está conformado por 6 personas designadas por el Jefe de Proyectos, dentro de las cuales cumplen una o más funciones dentro del mismo proyecto
* Las fechas de entrega fueron en común acuerdo con el jefe de proyectos y partes implicadas de manera que todos cuenten con disponibilidad para desarrollar y dar la retroalimentación pertinente.

Los roles se seccionaron de la siguiente forma:

**Tabla 1.**

*Descripción de los roles del equipo de trabajo y responsables.*

| **ROL** | **FUNCIÓN** | **ENCARGADO** |
| --- | --- | --- |
| **Desarrollador Backend** | Responsable de la programación del lado del servidor de la aplicación, lo que implica trabajar con bases de datos, servidores y la lógica del negocio. | Barbosa Motta, Branimir Rai |
| **Administrador de base de datos (DBA)** | Responsable de diseñar, implementar y mantener las bases de datos necesarias para el correcto funcionamiento de la aplicación. | Barbosa Motta, Branimir Rai |
| **Scrum Master** | Encargado de liderar el equipo de desarrollo de software, asegurando que se siga la metodología Agile y se cumplan los objetivos establecidos en el proyecto. | Coronel Vilca, Brisa Valeria |
| **Diseñador UX** | Responsable de diseñar la experiencia de usuario de la aplicación, trabajando en la interfaz gráfica y la usabilidad. | Jauregui Diaz, Yajahira Ysabel |
| **Desarrollador Frontend** | Responsable de programar el lado del usuario y de la aplicación, lo que implica trabajar con HTML, CSS y JavaScript para crear la interfaz de usuario. | Matos Ramos, Franco Antonio |
| **Jefe de proyecto** | Responsable de liderar el proyecto en su totalidad, estableciendo los objetivos, coordinando el equipo y asegurando que se cumplan los plazos. | Mayor Coloma, Fabrizio |
| **Arquitecto de software** | Encargado de diseñar la estructura del software, estableciendo las bases para el desarrollo del mismo y asegurando que sea escalable y sostenible en el tiempo. | Mayor Coloma, Fabrizio |
| **Analista QA** | Responsable de garantizar la calidad del software en todas las etapas del proceso de desarrollo. | Valega Vidarte, Renzo Omar |
| **Tester** | Verifica que el software cumpla con los requisitos especificados, asegurando su correcto funcionamiento, rendimiento, seguridad y usabilidad. | Valega Vidarte, Renzo Omar |

*Nota.* Elaboración propia.

Los criterios que se deben de cumplir cuando se culminan y aprueban actividades, es decir, en cuanto a funcionalidades sin realizar tests, puesto que estas se realizan en su semana correspondiente, son:

* **Funcionalidad completa:** Debe cumplir con los requisitos funcionales definidos en la reunión de planificación del sprint.
* **Integración:** Debe haber sido integrada en la rama principal del código y debe estar disponible para su uso en el producto final.
* **Documentación:** Debe haber creado y actualizado la documentación necesaria para el usuario final y para el equipo de desarrollo.
* **Diseño y código de calidad:** Deben cumplir con los estándares de calidad establecidos por el equipo de desarrollo.

Los miembros del equipo y partes implicadas deben de estar de acuerdo con los criterios. De esta manera, se garantiza que el trabajo realizado cumpla con los estándares de calidad y funcionalidad establecidos por el equipo y partes implicadas.

## **Definición de Requerimientos**

En base al alcance esperado del proyecto se determinaron las funciones clave y requerimientos con los que debe de cumplir, estos a su vez tienen que satisfacer las siguientes especificaciones del programa:

* **Lenguaje de programación:** Python 3.9
* **Framework web:** React 18.2.0
* **Motor de base de datos:** Postgres 9.6
* **Servidor web:** Gunicorn 20.1.0
* **Frontend:** React 18.2.0
* **Backend:** FastAPI 0.95.1

Asimismo, se distinguen tres tipos de usuarios:

* **Usuario:** Acceso total a la navegación de la plataforma.
* **Administrador:** Acceso al sitio administrativo de la plataforma para la gestión del contenido (información y datos) con el fin de realizar el seguimiento de los tickets y demás apartados de la plataforma.

De esta forma, se procede a enlistar los requerimientos obtenidos para el desarrollo de la plataforma web.

**Tabla 2.**

*Lista de requerimientos de la plataforma.*

| **ID** | **REQUERIMIENTOS** | **NOMENCLATURA** |
| --- | --- | --- |
| 01 | Mostrar el número de tickets | GT-ER\_01.DOCX |
| 02 | Notificar cuando no haya tickets | GT-ER\_02.DOCX |
| 03 | Historial de comidas ofrecidas | GT-ER\_03.DOCX |
| 04 | Calcular tiempo en llegar a 0 tickets | GT-ER\_04.DOCX |
| 05 | Calcular tiempo en llegar al comedor | GT-ER\_05.DOCX |
| 06 | Mostrar información nutricial comida | GT-ER\_06.DOCX |

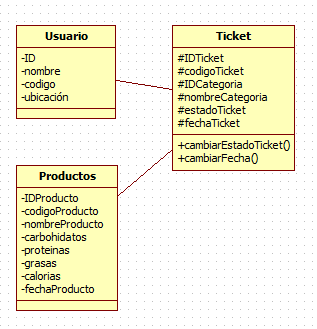
*Nota.* Elaboración propia.

## **Diseño**

De acuerdo con el enlistamiento de requisitos y funcionalidades iniciales se elaboró el siguiente diagrama de clases realizado en StarUML.

**Figura 3.**

*Modelo preliminar de las funcionalidades del sistema.*



*Nota.* Elaboración propia.

En el desarrollo del proyecto se añadirán y editarán estas clases, atributos o métodos en función con lo que se necesita en el programa. Asimismo, la base de datos toma un puesto importante dentro del programa, pero este no fue colocado en el diagrama.

De acuerdo al diseño frontend, solo se realizaron prototipos; posteriormente se implementarán junto al backend para las respectivas funcionalidades.

## **Desarrollo por Sprints**

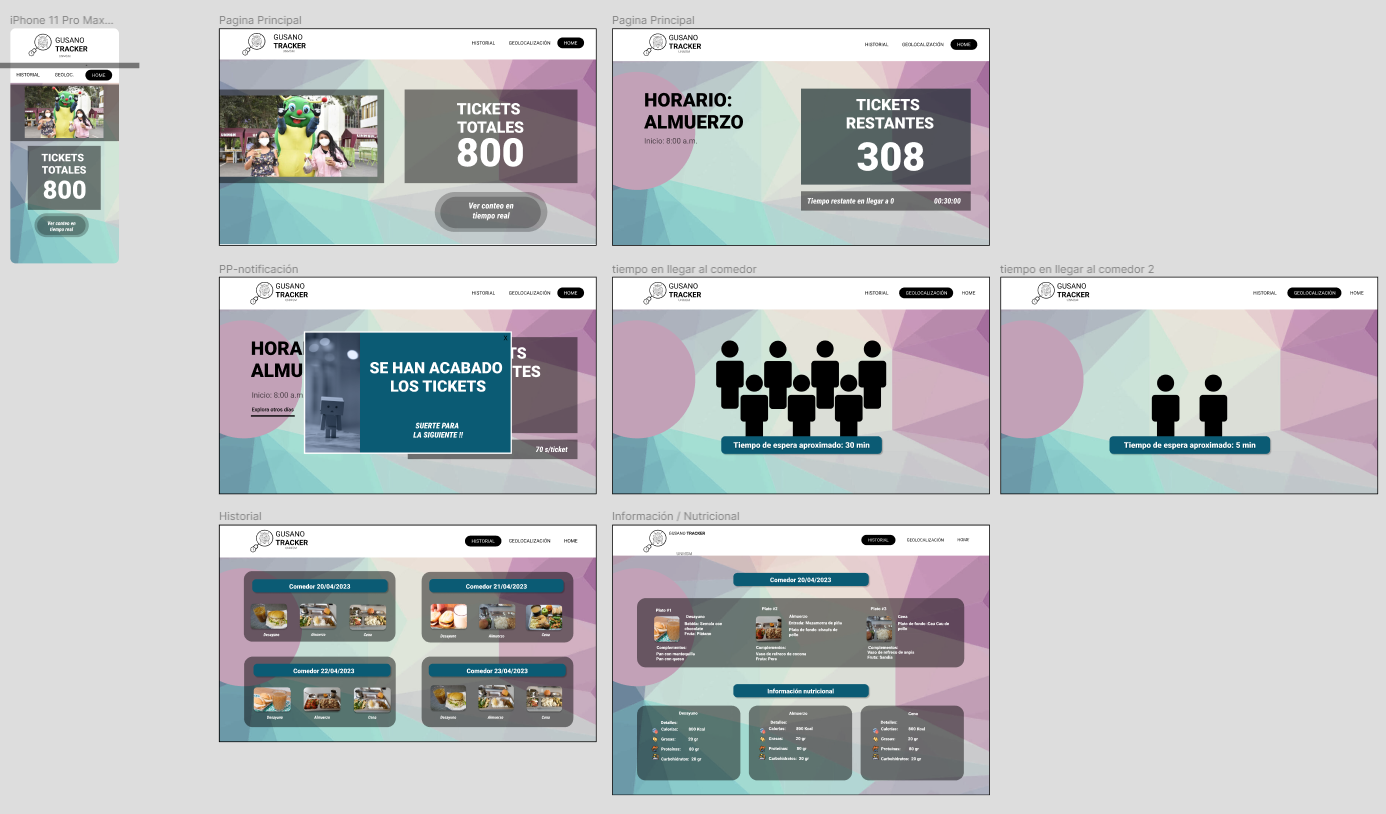
Los sprints se utilizaron para el desarrollo y manejo de diferentes tipos de documentos: la especificación de requisitos, estilos para la web, diseño de la base de datos, la arquitectura y diseño del software. Se adjuntan estos documentos en la sección de “Desarrollo” dentro del repositorio de la consultora de software, Macro System, en específico colocados en la carpeta “GT” por siglas del proyecto. Además que se referenciaron en cada ítem que aborda los sprints.

### **Sprint 1**

Conforme con el cronograma para la planificación de actividades “GT-CP”, se realizó la especificación de requisitos como el diseño de prototipos realizado en Figma. Asimismo, se establecieron los estilos para la web, diseño de base de datos, arquitectura y diseño del software.

**Figura 3.**

*Sprint 01- Prototipado para la plataforma web.*



*Nota.* Elaboración propia.

El primer sprint se centra en la realización y definición detallada de los requerimientos, junto con ello se establecieron los prototipos de pantalla para la plataforma web. Se contempló que cada uno de estos presenten fácil flujo de navegación como pertinente información para ser mostrada en cada una de las páginas a desarrollar. Aquel trabajo también fue representado en un documento especificando los elementos gráficos usados.

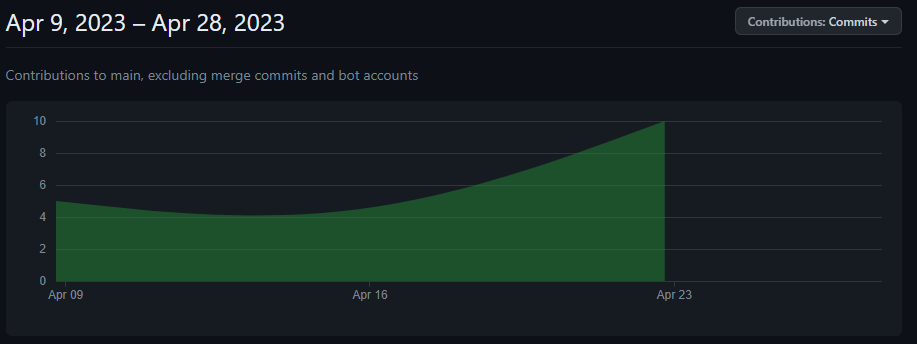
Por otro lado, el documento preliminar del diseño de la base de datos se destaca como un documento importante dado que se establecen las rutas que se deben seguir para el siguiente sprint (dado que conlleva codificación).

Por último, no hubo ningún incumplimiento con respecto a las fechas establecidas en el cronograma, aquello indica gran responsabilidad, experiencia y trabajo en conjunto por parte del equipo.

A continuación, se presentan las gráficas obtenidas por el repositorio de Github hasta el día 28 de abril del 2023.

**Figura 4.**

*Grado de actividad en el repositorio del 09-23 de abril.*



*Nota.* Todos los integrantes del equipo estuvieron involucrados.

**Figura 5.**

*Grado de actividad en el repositorio por integrante.*



Nota. El integrante “yjhr21” es representado por Jauregui Diaz, Yajahira Ysabel y el integrante “klatort” es representado por Mayor Coloma, Fabrizio.

Por ende podemos concluir que se llevaron éxito todas y cada una de las actividades hasta el 28 de abril del 2023.

# **Conclusiones**

En el primer sprint del proyecto “Sistema de Trackeo de Tickets en tiempo real del Comedor de UNMSM (GT)” se observó gran compromiso de parte de los colaboradores en relación con las actividades fechadas. Todos los avances se convierten y se acoplan para la realización final del producto con el fin de mantener la calidad con la que cuenta Macro System.

Se puede indicar que se logró cumplir con la gran mayoría de los objetivos específicos, con respecto a eso, se concluye que el trabajo realizado hasta el momento fue enfocado siguiendo las directrices anteriormente mencionadas.

Para finalizar, la implementación de la metodología híbrida (cascada y scrum) refleja que los responsables, actualmente, entienden y comprenden cómo y qué flujo deben de seguir. El ritmo de trabajo fue impecable, lo cual podría conllevar a una mejor adaptación y asimilación de la misma para entregas más rápidas permitiendo una mejora de la productividad.

# **Referencias**

*ALIMENTACIÓN*. (2016, enero 18). OFICINA GENERAL DE BIENESTAR UNIVERSITARIO. <https://ogbu.unmsm.edu.pe/index.php/alimentacion/>

De, N. (s/f). *La Guía Definitiva de Scrum: Las Reglas del Juego*. Scrumguides.org. <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf>

Schwaber, K. & Sutherland, J. (2020). La Guía de Scrum. Scrumguides.org. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>

Sommerville, I. (2010). *Software Engineering* (7a ed.). Addison-Wesley Educational