

# Sprint Retrospective

---



Arriaza Arriaza, Daniel	Morato Navarro, Juan Carlos
Barrera García, Ismael	Morato Navarro, Olegario
Calero López, Marina	Robles Russo, Eduardo
Galeano de Paz, Guillermo	Roldán García, Miguel Ángel
González Castellero, Rafael	Romero González, Juan
Márquez Sierra, María	Urquijo Martínez, Álvaro
Márquez Soldán, María	

Grupo	4	Entregable	S1
Repositorio	<a href="https://github.com/Cohabify/Cohabify">https://github.com/Cohabify/Cohabify</a>		

Versión	Cambios	Autores
V1.0	Creación del documento y contenido	Juan Romero González
V1.1	Modificación completa de la estructura y contenido.	Olegario Morato Navarro / Juan Carlos Morato Navarro

## Tabla de contenidos

Resumen ejecutivo.....	3
<b>1. Retrospectiva.....</b>	<b>3</b>
1.1 Semana 1.....	3
1.1.1 Aspectos positivos.....	3
1.1.2 Aspectos a mejorar.....	4
1.1.3 Acciones correctivas.....	4
1.2 Semana 2.....	5
1.2.1 Aspectos positivos.....	5
1.2.2 Aspectos a mejorar.....	6
1.2.3 Acciones correctivas.....	7
<b>2. Conclusiones.....</b>	<b>8</b>
2.1 Conclusiones Sprint 1.....	8

## Resumen ejecutivo

En el presente documento, se reunirán todos los aspectos tratados durante las diferentes reuniones de retrospectiva realizadas a lo largo de los sprints. El objetivo de la retrospectiva es tratar de resumir cómo se ha trabajado, qué es lo que se ha hecho bien, que se ha hecho mal, y cómo mejorar aquello que se ha hecho mal. Se separará el análisis del sprint en 2 semanas, ya que hemos tenido 2 retrospectivas.

# 1. Retrospectiva

## 1.1 Semana 1

Durante la primera semana se establecieron tareas relacionadas con la preparación del entorno y modelo de datos de la aplicación de Cohabify. Además se aprovechó para incluir tareas sencillas que permitieran al equipo adaptarse a las tecnologías que se usarían durante el resto del desarrollo.

A continuación se muestran de forma resumida los aspectos positivos, a mejorar y acciones correctivas tomadas durante la primera sesión de retrospectiva a mitad del Sprint 1.

### 1.1.1 Aspectos positivos

Aspectos positivos	Descripción
Las tareas se han realizado a un ritmo correcto	En aspectos generales, se han completado las tareas establecidas para la semana.
Ha funcionado muy bien la división en subgrupos	Los subgrupos G1 y G2 optaron por un <b>Pair Programming</b> para completar las tareas.
Ha sido muy útil la extensión Live Share de Visual Studio	El equipo destaca la utilidad de Live Share para aplicar Pair Programming.
Las revisiones de tareas han sido rápidas	Las tareas no han estado más de un día en estado de revisión.

### 1.1.2 Aspectos a mejorar

Aspectos negativos	Descripción
<b>Organización y selección de tareas</b>	Se asignaron tareas bloqueantes a miembros que tenían muy poca disponibilidad.
<b>Priorización de tareas según su urgencia</b>	Existían tareas muy importantes que no se realizaron tarde por una mala priorización de las mismas.
<b>Tareas que se han dejado para el fin de semana</b>	Este punto hace referencia al bloqueo provocado por asignar tareas a miembros con poca disponibilidad durante la semana.
<b>Mejorar la responsabilidad por parte de los Group Manager</b>	Uno de los Group Manager estuvo ausente durante la semana y su equipo tuvo que autogestionarse realizando las tareas propias de su Group Manager.
<b>Mejorar la comunicación entre los grupos.</b>	Aunque existía una buena comunicación dentro de los subgrupos, se desconocía el estado del resto de grupos.
<b>Carga de trabajo igualitaria</b>	La carga de trabajo estuvo descompensada entre algunos miembros del equipo.
<b>Backlog Incompleto</b>	El Product Backlog no se revisó correctamente y posteriormente se han detectado grandes problemas referentes a la cobertura de las funcionalidades definidas para la aplicación.

### 1.1.3 Acciones correctivas

Acciones correctivas	Descripción
<b>Intercambio de tareas si hay tareas bloqueantes sin hacer</b>	El equipo acordó poder reasignar tareas bloqueantes en caso de que el miembro asignado no pueda tener la tarea preparada a tiempo, para evitar los problemas provocados durante la semana anterior.
<b>Establecer fechas límite para las tareas</b>	Se establecieron fechas límites para evitar la tentativa de dejarlas para el último día.
<b>Informar al principio de semana de la disponibilidad de cada miembro</b>	Con el objetivo de repartir aquellas tareas prioritarias para los miembros con mayor

	disponibilidad.
<b>Utilizar el calendario Niko Niko</b>	Para poder saber en todo momento el estado de los miembros del equipo.
<b>Implementar correctamente la política de gestión de issues</b>	Para llevar un control del estado de las tareas de una manera más adecuada.

## 1.2 Semana 2

Durante la segunda semana se implementó prácticamente el core de la aplicación por completo, completando el Sprint Backlog en su totalidad. Aunque la comunicación ha mejorado, siguen existiendo problemas con uno de los Group Manager y su ausencia. Esto ha desencadenado en la dimisión como Group Manager del miembro para dejar a un candidato con más disponibilidad, algo importante para gestionar al equipo.

A continuación se muestran de forma resumida los aspectos positivos, a mejorar y acciones correctivas tomadas durante la última sesión de retrospectiva del Sprint 1.

### 1.2.1 Aspectos positivos

Aspectos positivos	Descripción
<b>El reparto de responsabilidades contribuye a eliminar el caos.</b>	Se ha asignado como responsabilidades parte del feedback a los miembros del grupo para asegurar que se responde a este y se mejora la calidad del proyecto.
<b>El establecimiento de un equipo de presentaciones.</b>	Asignar la presentación a dos miembros del equipo de forma fija permite dejar esta responsabilidad claramente asignada de cara al resto de semanas de trabajo y permite mejorar la calidad de exposición.
<b>Buena revisión de pull request en general</b>	Se han revisado las pull request de manera más exhaustiva para evitar problemas de código.
<b>Repartir responsabilidades de forma fija al principio de semana</b>	Esto ayuda a la organización del equipo, ya que se sabe en todo momento qué persona es responsable de cada tarea.

<b>Pair programming efectivo</b>	Gracias al pair programming, el equipo puede aprender de los demás compañeros mientras se trabaja en las tareas.
----------------------------------	------------------------------------------------------------------------------------------------------------------

### 1.2.2 Aspectos a mejorar

<b>Aspectos negativos</b>	<b>Descripción</b>
<b>Se han dejado tareas para el fin de semana</b>	Debido a la disponibilidad del equipo esta semana, se han dejado tareas para el fin de semana.
<b>Modelo de datos inestable</b>	Se ha seguido modificando el modelo de datos para llegar a un consenso.
<b>Documentación</b>	Se está descuidando la documentación y dejándose para el final.
<b>Tener una idea clara del producto core y sus componentes.</b>	Se debe tener una idea clara del producto para evitar confusiones sobre las decisiones de diseño y otros aspectos.
<b>Llevar a cabo un seguimiento del calendario Niko Niko</b>	Es importante rellenar el calendario niko niko para poder solucionar problemas en el equipo.
<b>Procurar que todas las vistas tengan un estilo homogéneo.</b>	Es necesario seguir un estilo homogéneo y acorde con los mockups para mantener la calidad visual en el proyecto.
<b>Tareas que no se han definido claramente.</b>	Existen tareas confusas que han retrasado el flujo de trabajo del equipo.
<b>Cambios en aspectos importantes de la aplicación que no se han notificado.</b>	Esto ha causado problemas en aquellas tareas que tenían dependencias con las modificaciones.
<b>No asignar backend y frontend a la misma persona.</b>	Esto ha causado conflictos al unir ambas tareas en el desarrollo
<b>Priorizar lo funcional.</b>	No parar el flujo de trabajo tratando de mejorar el aspecto de la aplicación y enfocarse primero en obtener la funcionalidad

### 1.2.3 Acciones correctivas

Acciones correctivas	Descripción
<b>Fijar el horario de las reuniones de retrospectiva a los domingos a las 12 de la mañana.</b>	Como respuesta a la problemática de juntar al equipo para las retrospectivas, se ha fijado un horario al que se comprometen todos los miembros para estar disponibles.
<b>Buscar información si en algún momento una tarea se complica</b>	Se propone no hacer uso directamente de la IA para resolver problemas de programación pues pueden equivocarse y hacer perder el tiempo al miembro implicado en la tarea.
<b>Corregir los estilos para homogeneizar el aspecto de la aplicación</b>	Se revisarán las pantallas creadas para corregir los estilos y mantener homogeneidad en la aplicación.
<b>Refactorizar el código generado durante el sprint 1.</b>	Es necesario refactorizar el código para evitar una deuda técnica mayor.
<b>Llevar la documentación de forma paralela con las demás tareas.</b>	Como respuesta al problema de documentos realizados los fines de semana, aquellos documentos que dependan del desarrollo estarán siempre actualizados de forma paralela a las demás tareas.
<b>Hacer un TODO list para los documentos y tareas que no son de desarrollo</b>	Dado que los documentos no estaban siendo realmente asignados a miembros, se propone asignar responsabilidades sobre los documentos ya que estaban provocando un gran desbalance en cuanto a carga de trabajo de los miembros.
<b>Cambiar la fecha del sprint de lunes a domingo</b>	Esta acción permitirá entregar con un día de antelación y facilitará el trabajo a los miembros encargados de realizar las presentaciones del martes.
<b>Seguir las pautas del team practices</b>	El equipo se compromete a seguir las Team Practices para obtener buenas métricas en Governify.

## 2. Conclusiones



## **2.1 Conclusiones Sprint 1**

Tras este primer sprint, el equipo puede concluir que se han aprendido valiosas lecciones que se pondrán en práctica de cara al segundo sprint. Aunque el trabajo en líneas generales haya ido bien, se espera que, mediante las medidas acordadas, la eficiencia y calidad del trabajo obtenido del sprint 2 aumente con respecto a este sprint.