**Retrospectiva**

* Tomamos la decisión de realizar 5 miniciclos.

1. Pruebas. Creamos pruebas para los métodos existentes.

* De este proceso desarrollamos 34 pruebas unitarias.

1. Desarrollo de pruebas, código (documentado) y diseño del nuevo constructor.
2. Desarrollo de pruebas, código (documentado) y diseño del método de mover los robots de la manera más efectiva.
3. Desarrollo de pruebas, código (documentación) y diseño del método de informar las tiendas y las veces que han sido saqueadas.
4. Desarrollo de pruebas, códigos (documentación) y diseño del método de informar los robots y las ganancias luego de cada movimiento.

* En esta parte incluimos lo de hacer que el robot con mayores ganancias parpadee.

1. Extra: Revisión y ajuste de algunos métodos.

* En esta parte ajustamos los colores reservados, agregamos el cambio de días a reboot, configuramos la forma de cambiar el color de las tiendas cuando son robadas.
* El ciclo dos está casi completo, sólo falta ajustar cómo se agregan los elementos que llegan en el nuevo constructor.
* Paula Díaz: 14 horas, Juan Pablo Vélez: 26 Horas
* El mayor logro fue cumplir con la funcionalidad de cada método nuevo y que respondieran correctamente a las pruebas y corregir algunos pequeños fallos del ciclo uno.
* El mayor problema fue encontrar algunas fallas para corregirlas, fallas de los métodos del ciclo uno. Las fallas no eran muy graves, solo fueron pequeños detalles que ajustamos para que la consistencia de los datos de las clases fuese correcta.
* El trabajo en equipo y para mejorar los resultados es tomar una actitud más diligente para no estar apurados a la hora de hacer entregas.
* Las prácticas XP empleadas fueron pair programming y Unit Test First. La más útil en este caso fue la UTF porque nos permitió diseñar las pruebas que nos guiaron luego para codificación de los nuevos métodos.
* La referencia que utilizamos fue el api de java.
* Utilizamos la guía de IA para el diseño de los métodos orderArray y orderArray2