**Documento de aplicación de Scrum como marco para la gestión del desarrollo del software Rayo Carro JGA.**

**Presentado a: Sandra Victoria Hurtado Gil**

**Jorge Iván Meza Martínez**

**Presentado por: Jhonathan González Aguirre**

**Asignatura: Procesos ágiles de desarrollo de software**

**Universidad Autónoma de Manizales.**

**Maestría en gestión y desarrollo de proyectos de software.**

**Manizales – caldas.**

**Abril de 2017**

Contenido

[1. Pila de producto: 3](#_Toc480150127)

[2. Planeación (pilas de sprint y tareas): 3](#_Toc480150128)

[Sprint # 1 3](#_Toc480150129)

[Sprint #2 4](#_Toc480150130)

[3. Seguimiento (cambios en tablero Kanban): 4](#_Toc480150131)

[4. Resultados de las retrospectivas: 5](#_Toc480150132)

[Sprint retrospective 14/03/2017 5](#_Toc480150133)

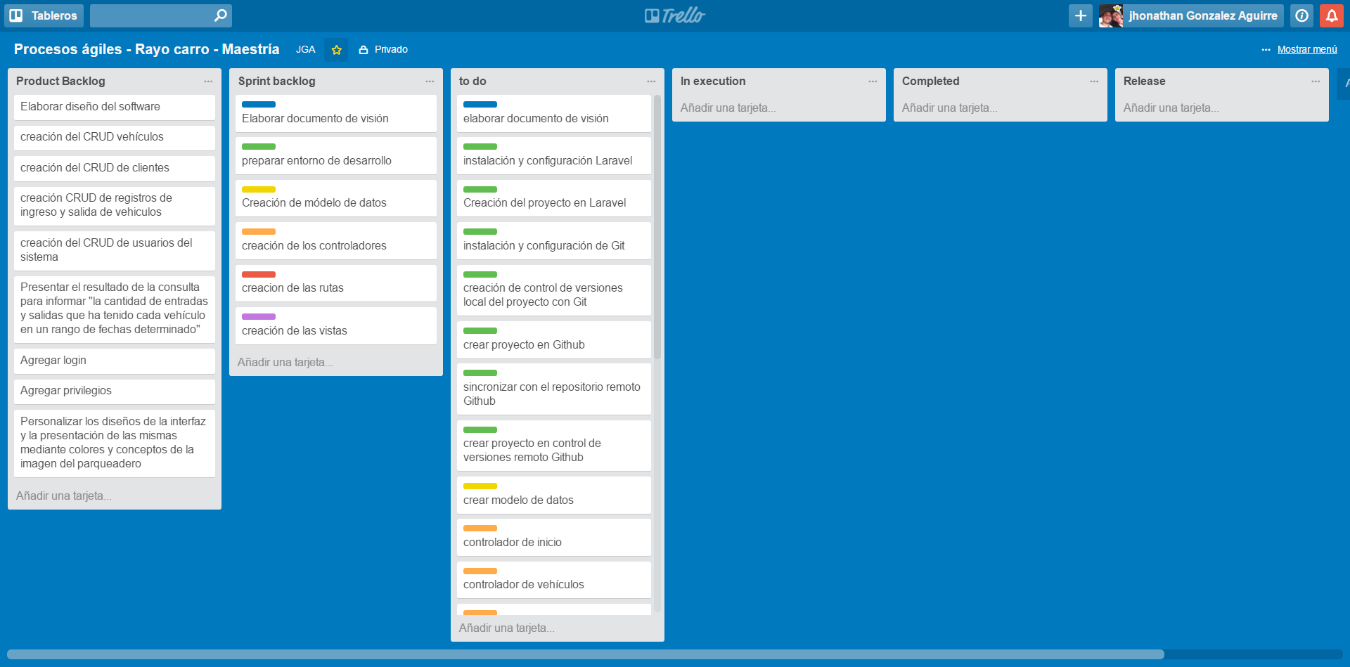
[Sprint retrospective 08/04/2017 6](#_Toc480150134)

# Pila de producto:

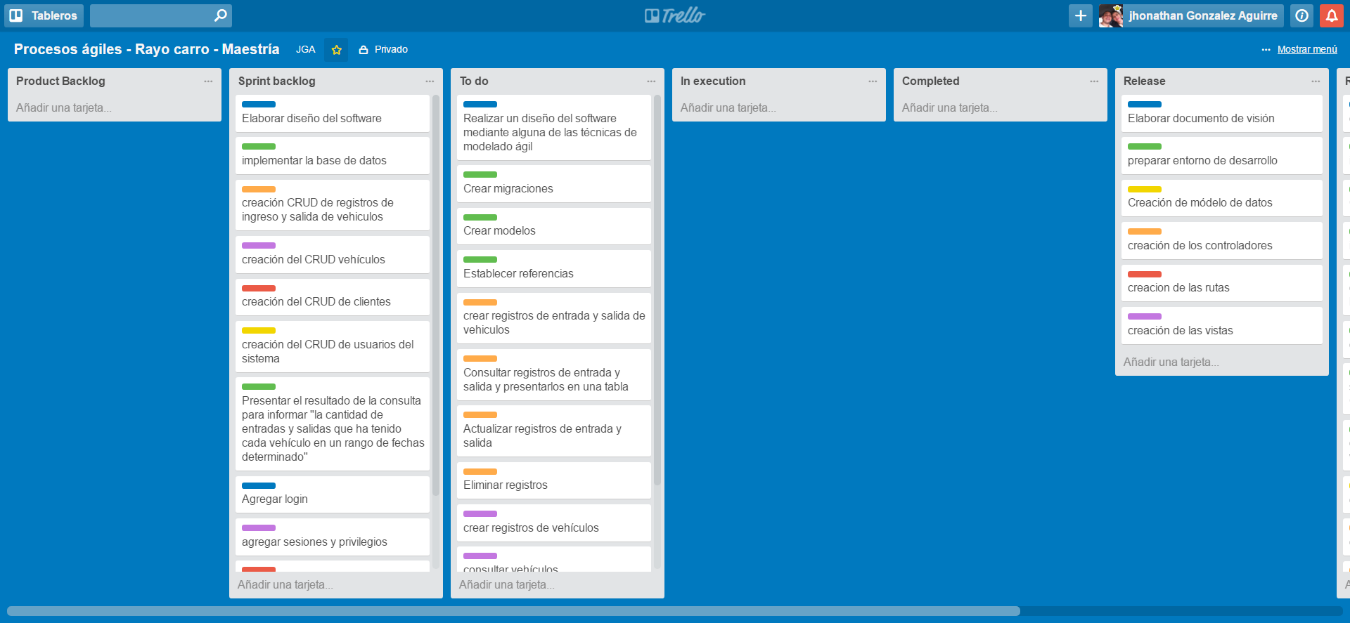
* 1. Preparar entorno de desarrollo
  2. Realizar diseño ágil
  3. Creación del modelo de datos
  4. Elaboración del documento de visión
  5. Creación de controladores
  6. Creación de las rutas
  7. Creación de las vistas
  8. Creación del CRUD de vehículos
  9. Creación del CRUD de clientes
  10. Creación del CRUD de registros de ingreso y salida de vehículos
  11. creación del CRUD de usuarios del sistema
  12. Presentar la cantidad de entradas y salidas que ha tenido cada vehículo en un rango de fechas determinado
  13. Agregar formulario para iniciar sesión
  14. Administrar sesiones y agregar privilegios
  15. Personalizar los diseños de la interfaz y la presentación de las mismas mediante colores y conceptos de la imagen del parqueadero

# Planeación (pilas de sprint y tareas):

## Sprint # 1



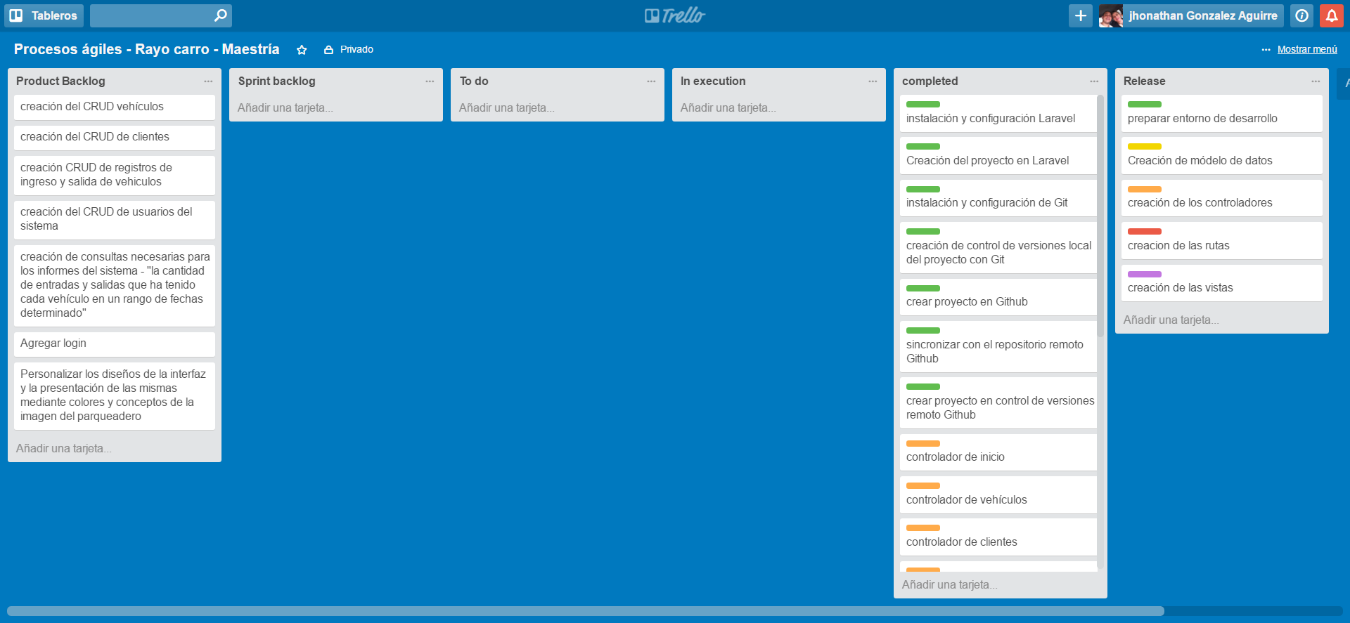
## Sprint #2

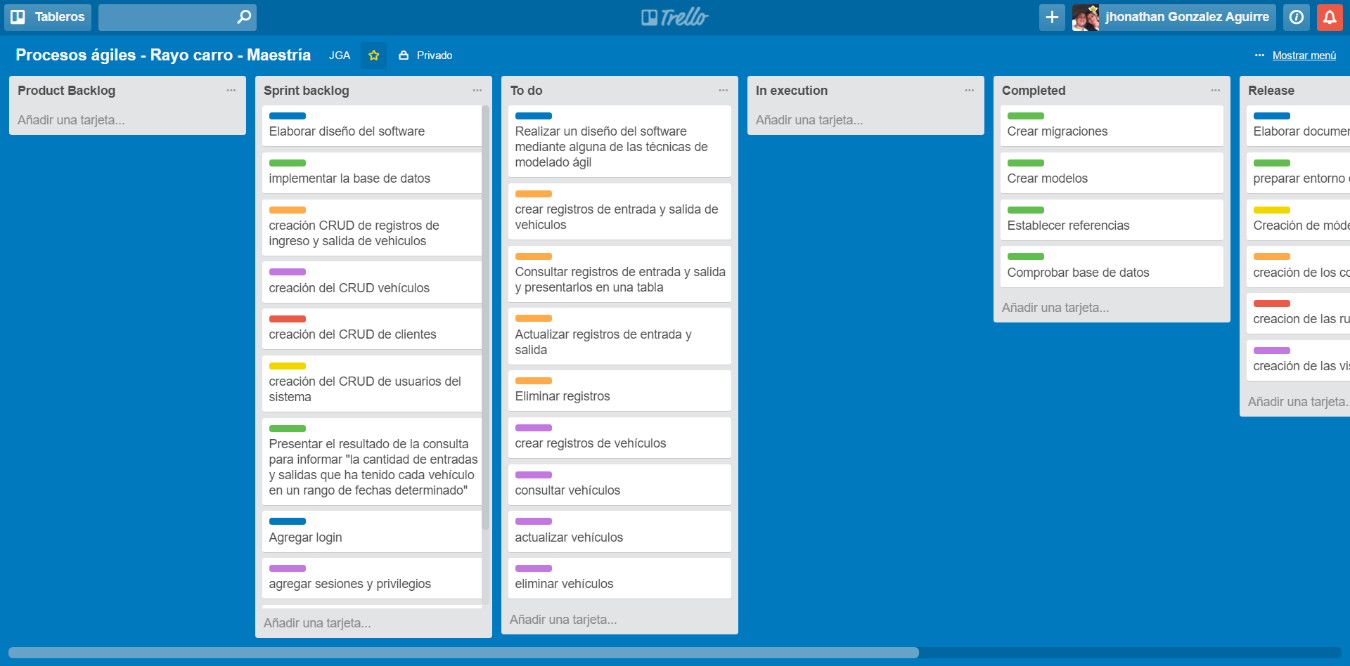


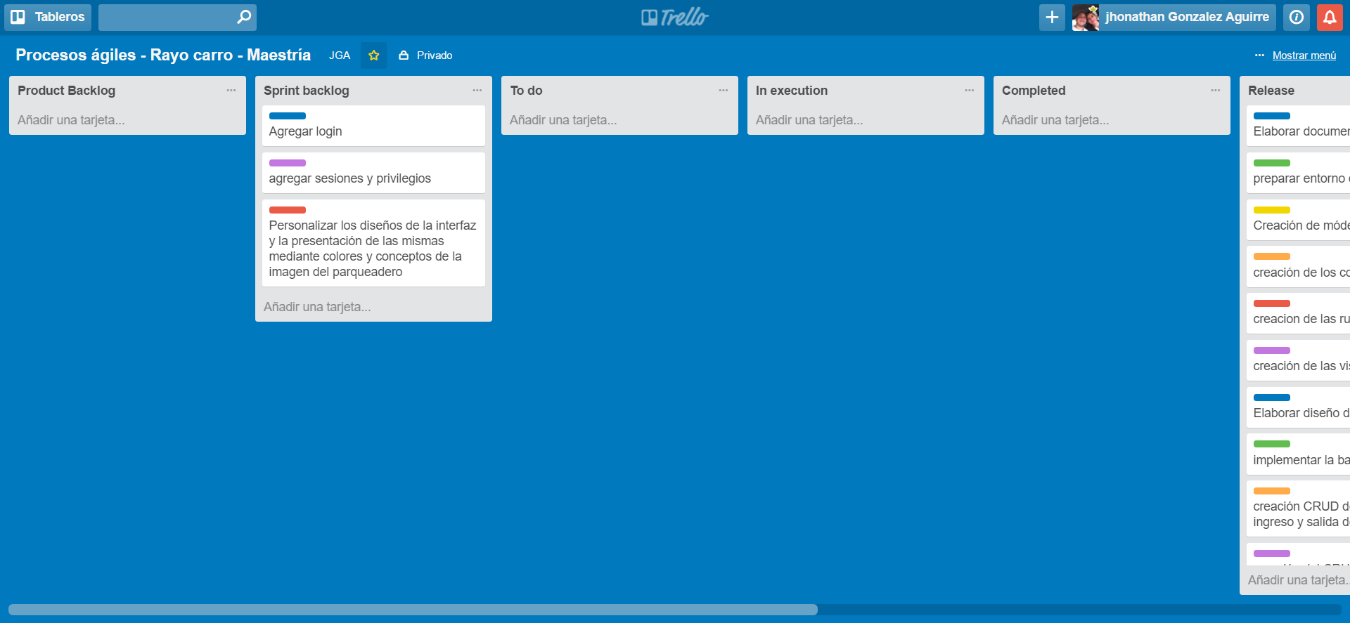
# Seguimiento (cambios en tablero Kanban):

En el siguiente enlace encuentra los *daily meeting* realizados durante el proyecto: <https://github.com/jonathanGonzalez/Laravel_procesos_agiles/tree/master/proceso/Daily%20meetings>

Otras imágenes evidencia del seguimiento son:

**Final sprint #1**

**Periodo Intermedio Sprint # 2**

**Final sprint #2**

# Resultados de las retrospectivas:

|  |
| --- |
| Sprint retrospective 14/03/2017 **Por mejorar en el proceso:**  **1.** se debe considerar no hacer tareas que no estén en el plan del día que puedan desviar la atención en lo verdaderamente importante en el caso de las vistas, se intentó implementar una plantilla y no tuvo resultado, lo que afecto el proceso y amplio el tiempo de trabajo considerado en el plan.  **2.** Falto elaborar una arquitectura simple del sistema que incluyera comprender las tecnologías antes de usarlas, conocer versiones y dependencias de una tecnología con otra, específicamente el proceso se vio afectado por no haber elaborado un borrador inicial de una arquitectura donde se definieran las versiones php y Laravel claramente. Al configurar laravel no se tuvo en cuenta las versiones y requerimientos, se intentó utilizar una versión de php que estaba en el sistema previamente lo que no funciono, posteriormente se investigaron los requisitos previos de instalación de laravel y se concluyó que debía instalarse otra versión de PHP. los requerimientos específicos encontrados en <https://laravel.com/docs/5.4>  **3.** Falto preguntar al "product owner" con claridad un requerimiento ambiguo en el enunciado:  Cuando se habla de "gestión básica de usuarios" la palabra "usuarios" se refiere a los clientes usuarios del parqueadero o a los usuarios del sistema (con privilegios).  **PARA FUTUROS SPRINT:** Para los siguientes sprint se debe priorizar las tareas de diseño por encima de la codificación, evita realizar reprocesos o perder tiempo en la codificación de funciones innecesarias. Un buen diseño genera un software más estable, mantenible y facilita las pruebas.  No hacer tareas que no estén planificadas, excepto una tarea que al no realizarse genere un impacto negativo y que no se haya identificado en la planificación. |

|  |
| --- |
| Sprint retrospective 08/04/2017 Posiblemente no es necesario un CRUD de clientes ya que agrega complejidad al usuario de forma que debe hacer un paso más antes del registro de una entrada/salida de un vehículo, debería ingresar una persona y un vehículo previamente. (La gravedad es baja, puede impactar en lo que espera el product owner)  No se validaron oportunamente los requerimientos, la entrega realizada asume que los requerimientos, principalmente el descrito en el párrafo anterior, están bien definidos.  Scrum define una participación activa del product owner y la responsabilidad del Scrum master de resolver dudas y dar garantías a los desarrolladores, falto articular correctamente los roles ya que no se presentaron las dudas al product owner (en el presente proyecto los docentes).  **PARA FUTUROS SPRINT:** Antes de ejecutar cualquier actividad se debe estar seguro de que el objetivo es claro y el resultado esperado está bien definido, no se debe codificar ninguna función sobre la cual exista alguna duda y la cual pueda tener impactos negativos en la arquitectura del sistema o cause la modificación amplia de la misma. |