

**Manual de Coordinación**

**Sistema de Diagnóstico y  
Agendamiento de Citas  
(SDAC)**

**Versión 1**

**Jairo Daniel Bautista Castro**

**Miguel de Oliveira Dias Gonçalves**

## Índice

<b>Ciclo de vida .....</b>	<b>2</b>
<b>Metodología de desarrollo.....</b>	<b>2</b>
<b>Recursos Software Desarrollo .....</b>	<b>2</b>
<b>Organización del equipo de trabajo (estructura, normas).....</b>	<b>2</b>
<b>Herramientas para comunicaciones en el equipo de trabajo .....</b>	<b>2</b>
<b>Relaciones con el cliente (entrevistas, reuniones, revisiones, ...).....</b>	<b>2</b>
<b>Estándares de documentación .....</b>	<b>3</b>
<b>Estándares de código.....</b>	<b>3</b>
<b>Control de versiones (método y herramientas) .....</b>	<b>3</b>
<b>Gestión de calidad (proceso y herramientas).....</b>	<b>3</b>

## **Ciclo de vida**

Seguiremos un ciclo de vida ágil, basado en los marcos de desarrollos Scrum y Kanban.

## **Metodología de desarrollo**

Seguiremos la metodología de desarrollo Extreme Programming. De esa forma, haremos integración y despliegue continuos; programación en pareja; desarrollo basado en pruebas, entre otros.

## **Recursos Software Desarrollo**

Usaremos el IDE Visual Studio Code tanto para desarrollar el back-end en Node.js; como también el front-end en Angular 2 y NativeScript.

Para las bases de datos, usaremos MongoDB y el IDE Robo 3T.

La herramienta de despliegue será Jenkins.

Usaremos Chef para orquestar las máquinas virtuales necesarias.

## **Organización del equipo de trabajo (estructura, normas)**

Una vez que se seguirá el marco de desarrollo Scrum, habrán 3 células de trabajo: una para las bases de datos; otra para el back-end; y una tercera para el front-end, con tamaño entre 3 a 9 personas. Cada célula tendrá un Scrum Master y un Product Owner, todos los restantes miembros de las células serán Developers.

Usaremos Trello para crear los tableros Kanban y gestionar las historias de usuario y el Product Backlog de Scrum.

## **Herramientas para comunicaciones en el equipo de trabajo**

Usaremos grupos en Slack para comunicación continua e intercambio de ficheros dentro del equipo; correo electrónico para comunicación escrita e intercambio de ficheros entre equipos; y Skype para videollamadas entre equipos.

## **Relaciones con el cliente (entrevistas, reuniones, revisiones, ...)**

En el fin de cada sprint, el cliente estará presente en el “Sprint Review” de modo a ser informado de los avances en el proyecto y dar su retroalimentación del proyecto a los equipos de desarrollo. El cliente tendrá

también acceso sólo de lectura a los tableros Kanban, historias de usuario y el Product Backlog creados en Trello por todos los equipos de trabajo.

### **Estándares de documentación**

Cada acta producida después de un “Sprint Review” del proyecto deberá tener la firma de, al menos, un representante del cliente e uno de cada equipo de trabajo.

### **Estándares de código**

Todos los comentarios del código deberán ser escritos en inglés. Deberá haber un comentario antes de cada “if”, “for”, “while” o similares.

### **Control de versiones (método y herramientas)**

Usaremos GitHub como herramienta principal de control de versiones. La rama “master” estará reservada al código final, esto es, más estable y probado; se subirá código a esa rama mediante “pull requests” que serán aceptados después de al menos 2 miembros del equipo lo revisaren. Para subir código en las restantes ramas no individuales, bastará que otro miembro del equipo revise el código.

### **Gestión de calidad (proceso y herramientas)**

El cliente tendrá acceso sólo de lectura a la rama “master” del repositorio Git y podrá a cualquier momento hacer sugerencias y proponer cambios en el código y funcionalidades, que serán analizadas por el Scrum Master y por el Product Owner.