TRABAJO PRÁCTICO ESPECIAL

INTRODUCCIÓN A LAS METODOLOGÍAS DE DESARROLLO DE SOFTWARE

GRUPO 4

PO: GUILLERMO RODRÍGUEZ

Viviana Ayelen Diaz Bailey, Beltran Peña, Ivan Samudio, Ezequiel Fernandez, Lautaro Morales, Lucia Blanco.

HERRAMIENTAS UTILIZADAS

- Grupo de whatsapp
- Jira Software
- Figma
- https://github.com
- Visual Studio Code
- XAMPP
- Hatjitsu

SPRINT GOAL

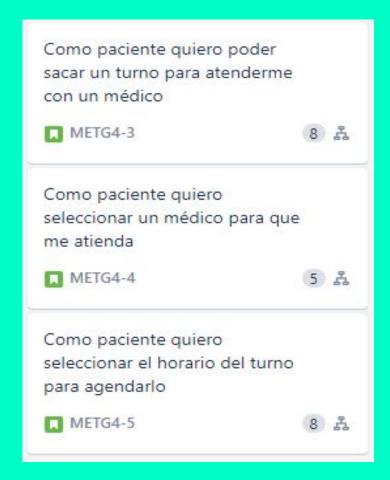
La startup TurnoFacil nos ha contactado para desarrollar un software que permita sacar turnos para médicos. Sus intenciones son contar con una aplicación web fácilmente instanciable.

Se nos solicitó implementar el poder visualizar los turnos disponibles de un médico siendo un paciente y confirmar un turno y ser notificado por email.

Planteadas las necesidades de esta empresa realizaremos este proyecto utilizando la metodología de trabajo Scrum.

USER STORIES

Se decidió cuáles eran las user stories que abarcaban el objetivo.



SUBTAREAS

Refinamiento: Se subdividió en otras más pequeñas para mejorar la delegación de responsabilidades.

METG4-26	Implementar filtro por especialidad y/o obra social
METG4-21	Crear interfaz grafica
METG4-27	Crear la tabla obra social en la base de datos
	Crear la tabla medico en la base de datos
METG4-28	Crear la tabla turno en la base de datos
METG4-37	Crear la tabla paciente en la base de datos
METG4-30	Crear la interfaz gráfica
En allerance accuse	Crear la interfaz gráfica Crear el mvc de médico
En allerance accuse	Crear el mvc de médico
METG4-31 METG4-32	Crear el mvc de médico
METG4-31 METG4-32 METG4-33	Crear el mvc de médico Mostrar los turnos del médico seleccionado

POKER PLANNING

• **Estimación:** participó todo el equipo pudiendo así saber cuánto esfuerzo llevaría cada user story (user story points).

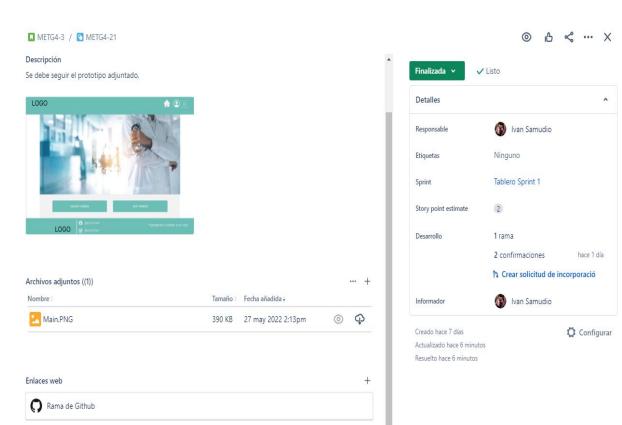
hatjitsu Room: 98505 Choose your estimate... Mountain Goat pack 13 20 40 100 ? 8 0 You are a voter You haven't estimated yet S Reset Reveal

Story point estimate

2

GITHUB

- Vinculamos cada subtarea con su respectiva rama mediante el id identificatorio de Jira.
- Además se agregó el enlace web de la url de la rama del repositorio.



DOCUMENTACIÓN

Incidencias:

- METG4-21: METG4-21: CHECK
- METG4-26: METG4-26: CHECK
- METG4-27: METG4-27: CHECK
- METG4-28:METG4-28: CHECK
- METG4-29: METG4-29: CHECK
- METG4-30: METG4-30: CHECK
- METG4-31: METG4-31: CHECK
- METG4-32:METG4-32: CHECK
- METG4-33:METG4-33: CHECK
- METG4-34: METG4-34: CHECK
- METG4-35:METG4-35: CHECK
- METG4-37: METG4-37: CHECK

Testeo de las incidencias del Sprint 1

Para que el testeo sea exitoso deberá cumplir con:

- Criterio de aceptación.
- No tener errores técnicos ni funcionales.

Si esto ocurre luego del nombre de la incidencia agregar:

- CHECK y pintarlo de amarillo. (Ej: METG4-1: CHECK).
- REHACER y pintarlo en color rojo seguido del por qué. (Ej: METG4-1:REHACER. Por qué no funciona tal cosa.)

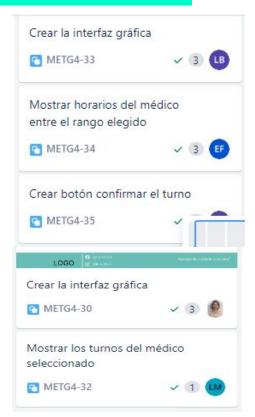
En caso de que deba rehacerse la próxima verificación será debajo de la anterior mencionando nuevamente su respectivo id.

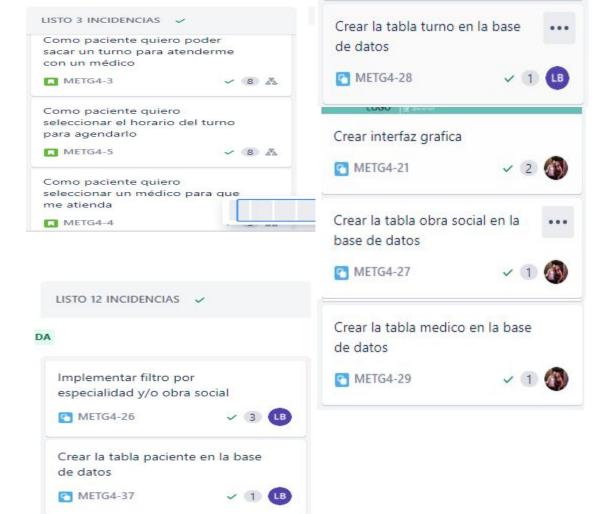
Ej: METG4-1:REHACER. Porque no cumple con el criterio de aceptación ya que..

METG4-1: CHECK / METG4-1: REHACER. Faltó el botón de...

POR HACER EN CURSO 3 INCIDENCIAS VERIFICAR LISTO 🗸

SPRINT 1 FINALIZADO









¿Qué deberíamos mejorar?

- Experiencia en github y jira.
- Mayor tiempo para el desarrollo.

Algunas propuestas para mejorar lo anteriormente planteado son:

- Definir la fecha de inicio del sprint con más anterioridad para darle más tiempo al desarrollo.
- Estudiar el funcionamiento de GitHub y Jira Software.



¿Que funcionó bien?

- Comunicación en el equipo.
- Buenas prácticas.

BIBLIOGRAFÍA

- https://confluence.atlassian.com/alldoc/atlassian-documentation-32243719.html
- https://github.com/jgraph/drawio/wiki/Getting-Support
- https://www.scrum.org/forum/scrum-forum







