Plataforma Historial de Equipos

Versión	Fecha Modificación
1	Octubre 19 2020

Visión de Producto

La **Plataforma Historial de Equipos**, es una herramienta donde el personal del laboratorio de informática (LabInfo) pertenecientes a la decanatura de Ingeniería de Sistemas de la Escuela Colombiana de Ingeniería Julio Garavito, pueden registrar los laboratorio, equipos y elementos de cada equipo, junto con las novedades que se realizan a cada uno de estos. El sistema, más allá de facilitar el registro de los equipos y novedades, es una valiosa base de conocimiento donde el personal del laboratorio, puede revisar el histórico de novedades que se le han realizado a cada elemento a través del tiempo durante todo su ciclo de vida útil. El personal administrativo del laboratorio puede crear laboratorios, equipos y elementos, asociar unos a otros y registrar novedades para cada uno; además de tener una variedad de reportes que les permitirá tener el control administrativo de estos implementos.

Detalles provistos por el *Stakeholder.*

El objetivo del sistema es permitir el registro y seguimiento de las novedades que han sido realizadas sobre los equipos de cómputo pertenecientes al Laboratorio de Informática. La plataforma de historial de equipos debe contar con una interfaz de usuario amigable con los usuarios.

Fechas Relevantes (Sprint)

Fecha	Actividad
Oct 20	Planeación Sprint Inicio Sprint 1
Nov 2	Revisión Sprint 1 Retrospectiva Sprint 1 Planeación Sprint 2
Nov 3	Inicio Sprint 2
Nov 16	Revisión Sprint 2 Retrospectiva Sprint 2 Planeación Sprint 3
Nov 17	Inicio Sprint 3
Nov 30	Revisión Sprint 3 Sustentaciones

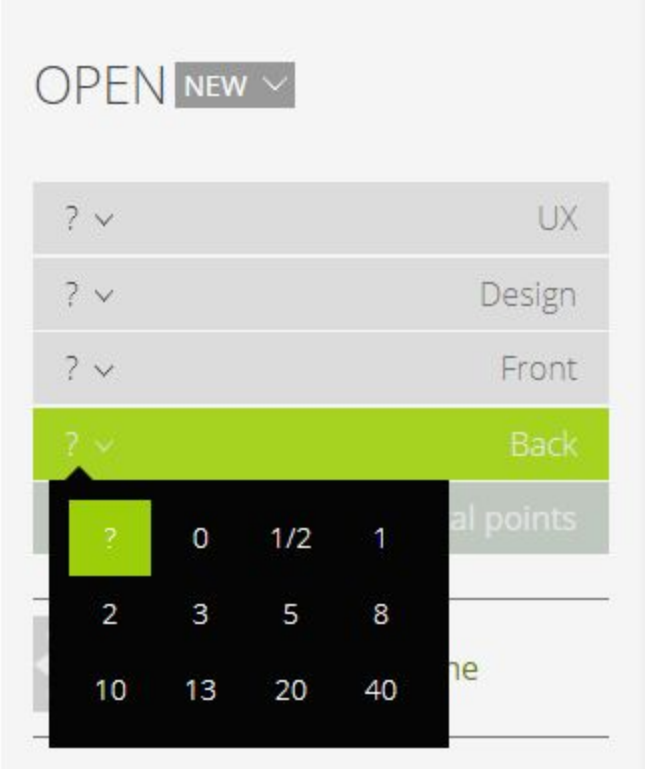
Actividades

1. Crear una organización en GitHUB para el proyecto.

2. Cada uno de los integrantes creará una cuenta en Taiga. Se debe importar la configuración y las historias de usuario provistas en la carpeta compartida.

Ponerle como nombre: 2020-2-PROYCVDS-NOMBRE_EQUIPO, compartirlo con los otros integrantes y permitir el acceso público.

3. Realizar la estimación, en 'puntos de historia' (planning poker) de las historias de usuario. Las historias de usuario ya se encuentran priorizadas de acuerdo con la visión de producto. La estimación de cada historia, debe realizarse en horas y quedar registrada en la misma herramienta.



4. Cuando se hayan definido las historias de usuario que van dentro del Sprint, en Taiga se deben agregar al respectivo Sprint, se debe realizar una planificación más detallada, indicando subtareas y responsables, y haciendo una estimación en horas de dichas tareas. En la herramienta, también hay un campo de estado para cada una de las historias de usuario que se está trabajando en el Sprint, es muy importante que la herramienta refleje el estado. Tenga en cuenta que, para cada Sprint, de acuerdo con los créditos del curso, el equipo contará con aproximadamente 21 horas por integrante (equipos de 4: aprox 80 horas).

5. El stack de tecnología del proyecto será el mismo que se ha trabajado en el curso

a lo largo del semestre. Sin embargo, se utilizará un esquema de base de datos POSTGRESQL, creado en Heroku. Las indicaciones de cómo crear dicha base de datos, y cómo configurar el acceso a la misma las puede encontrar acá:

https://github.com/PDSW-ECI/Heroku-Base-de-datos

Junto con un ejemplo de implementación completa acá:

https://github.com/PDSW-ECI/guice-mybatis-postrgres-demo

6. Configurar la conexión entre el proyecto de Taiga y el repositorio de Github.

Condiciones de entrega

1. Sin excepción, no se revisarán proyectos que no tengan configurado un entorno de entrega y despliegue continuos. La evaluación de las historias de usuario seleccionadas se realizará con lo que haya publicado en 'la nube'.
2. No se revisarán fuentes en un medio diferente al del sistema de control de versiones. En el mismo debe quedar evidencia del aporte realizado por cada usuario. Por favor ESTANDARIZAR LOS NOMBRES DE USUARIO DE GIT, para evitar tener reportes de avance inconsistentes.
3. Todos el software utilizado no debe tener costo.

Condiciones de entrega final (20% del examen final)

Para la revisión final (tercer Sprint), en el archivo README.md (que debe estar ubicado en la raíz del repositorio de GitHub) y haciendo uso del formato Markdown (https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet), incuír:

- Nombre del proyecto
- Período académico, nombre del curso, nombre de los integrantes, nombre del profesor, roles asignados (no olvidar que el profesor tuvo el rol de 'dueño de producto').
- Descripción del producto.
    - Descripción general.
    - Manual de usuario.
    - Imágenes y descripción de las funcionalidades más importantes.
- Arquitectura y Diseño detallado:
    - Modelo E-R.
    - Diagrama de clases (hacerlo mediante ingeniería inversa)
    - Descripción de la arquitectura (capas) y del Stack de tecnologías utilizado (PrimeFaces, Guice, QuickTheories, PostgreSQL).
    - Enlace a la aplicación en Heroku.
    - Enlace al sistema de integración continua.
- Descripción del proceso:

- Integrantes.
- Breve descripción de la Metodología.
- Enlace a Taiga (hacer público el Backlog).
- Generar el 'release-burndown chart' del proyecto, e indicar los puntos de historia realizados y los faltantes.
- Para cada Sprint:
  - Imagen del 'sprint-backlog'
  - Imagen del 'sprint-burndown chart' (sacado del sprint-backlog anterior), y una descripción breve de los problemas encontrados y mejoras realizadas al proceso.
- Reporte de pruebas y de cubrimiento de las mismas (sólo la foto del reporte principal). Para la cobertura, pueden usar los plugins disponibles (EclEmma, Jacoco, etc.)
- Reporte de análisis estático de código. Se pueden usar las mismas herramientas trabajadas en los laboratorios.