****

**Metodologías Ágiles en el Desarrollo de Software**

**Nombre del grupo: Equipo Lambda**

**Integrantes:**

* Carranza Exequiel 60848
* Cruz, Karen Yanina 61539
* Ricagno, Carolina 70374
* Velasco, Melisa Paola 58496
* Tagliaferri, Stefano 62890

**Proyecto:** Software de Pasantías

**Entregable 1: Plan de proyecto**

**Proyecto de software de Pasantías**

**Objetivo:** brindar información para que las empresas puedan gestionar sus pasantías vigentes y que los alumnos de la Universidad Tecnológica Nacional puedan inscribirse a las mismas.

**Alcance:** Desde que una empresa tiene la intención de publicar pasantías vigentes hasta que el alumno se inscribe provisoriamente a la misma.

**Tecnologías**

**Back-end:**

* C# - Visual Studio: Para desarrollar la aplicación web utilizaremos el lenguaje C#, dentro del IDE Visual Studio.
* MySQL: Utilizaremos esta herramienta para definir la base de datos que implementaremos en los diversos sprints.

**Front-end**

* HTML: Es un lenguaje que se utiliza para el desarrollo de páginas de Internet. Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos.
* CSS3: CSS es un lenguaje para definir el estilo o la apariencia de las páginas web, escritas con HTML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas.
* Bootstrap: Es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS, con la particularidad de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice.
* Angular: Es un framework JavaScript, gratuito y Open Source, creado por Google y destinado a facilitar la creación de aplicaciones web modernas de tipo SPA.

**Herramientas:**

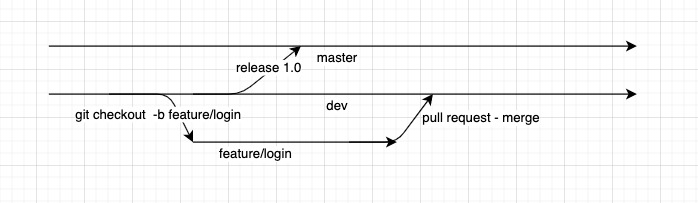
La herramienta que utilizaremos para gestionar el proyecto utilizando una orientación ágil será la plataforma proporcionada por Microsoft llamada **Azure DevOps**, donde se manejan diferentes usuarios y una completa definición de User Stories, Sprints, asignación de tareas y métricas. [Link a Azure DevOps](https://dev.azure.com/EquipoLambda/EquipoLambda-SoftPasantias)

Dicha plataforma trabaja conjuntamente con **GitHub**, que nos garantiza el control de versiones del producto y, así lograr la integración del trabajo de todo el equipo.

[Link a GitHub](https://github.com/exe0708/equipoLambdaBackEnd.git) (backend)

[Link a GitHub](https://github.com/caroric/equipoLambda.git) (frontend)

**Criterios de manejo del repositorio**

****

Vamos a manejar una rama **master** de la cual desprende una rama **dev (desarrollo en backend y develop en frontend)** de la cual desprende una rama por US la cual una vez finalizada la funcionalidad se hace el pull request -merge a **desarrollo**, la única rama que realiza el pull request -merge a **master** es la de **desarrollo** cada vez que haya una release del producto.

**Ideas del producto:**

**Roles:**

* **Scrum Master:** Stefano será el encargado de liderar al equipo para que cumplan las reglas y procesos de la metodología y para que el proyecto marche en buenos términos.
* **Product Owner:** Juan Pablo Bruno será el encargado de representar al cliente y a los que utilizan el software.
* **Team:** Exequiel Carranza, Karen Cruz, Carolina Ricagno y Melisa Velasco conformarán el equipo ágil que realizará de forma alternada las tareas del desarrollo y testing para cada entregable.

**Métricas**

* **Velocidad.** Cantidad de story points realizados por sprint.
* **Capacidad.** Cantidad de horas trabajadas por sprint.

**Criterio de DONE**

* Testing cruzado
  + Se realizará un testing a todos los campos obligatorios.(Crítico)
  + Se realizará un testing a todos los campos opcionales.(Bajo)
  + Se realizará un testing de la funcionalidad del camino óptimo de la US(Crítico)
  + Se realizará un testing para el camino de falla de la US(Medio)
  + Se realizará testing de visibilidad de componentes de pantalla en base a comportamiento.(Bajo)
  + Se realizará un testing de los bugs corregidos(Medio)
  + Se realizará un testing de seguimiento de estados.(Medio)
  + Se realizará testing de verificación de datos almacenados(Medio/Crítico)
* Testing de aceptación por parte del product owner.
* Definir tablas de nivel de aceptación.
* Cumple con las validaciones.
* Cumple con el diseño.
* Cumple con la lógica de negocio.

**Roles**

* Empresa
* Alumno
* Encargado de Pasantías

**Casos de Uso:**

**Publicación de las pasantías.**

* Registrar Empresa
* Registrar pedido pasantías

**Control de pasantías a publicar.**

* Ver Pedidos de Solicitud de Pasantía de empresas
* Aceptar Solicitud de Pasantía de la empresa
* Cancelar Solicitud de Pasantías de la empresa
* Ver Empresa
* Ver Pedido Pasantía

**Inscripción de alumnos a las pasantías.**

* Aceptar Pedido de Pasante
* Registrar alumno
* Inscribir a pedido de pasantías
* Ver Pasantías Activas
* Buscar Pasantías
* Filtrar pasantías
* Mostrar datos pasantía
* Mostrar pasantías

**Selección/Entrevista de alumnos.**

* Ver Alumno
* Ver alumnos inscriptos para entrevista
* Ver alumnos seleccionados pasa pasantía
* Registrar pedido de Pasantías del alumno para entrevista

**Aceptación pasante.**

* Registrar aceptación de alumno para Pasantías
* Registrar plan de Pasantías

**Generación de convenio.**

* Generar Convenio Particular
* Aceptar Pasante

**Separado por roles:**

**Empresa**

* Registrar pedido pasantías
* Registrar Empresa
* Ver alumnos inscriptos para entrevista
* Ver alumnos seleccionados pasa pasantía
* Registrar pedido de Pasantías del alumno para entrevista
* Registrar aceptación de alumno para Pasantías
* Registrar plan de Pasantías

**Alumno**

* Registrar alumno
* Inscribir a pedido de pasantías
* Ver Pasantías Activas
* Buscar Pasantías

**Encargado de Pasantías**

* Ver Pedidos de Solicitud de Pasantía de empresas
* Aceptar Solicitud de Pasantía de la empresa
* Cancelar Solicitud de Pasantías de la empresa
* Aceptar Pedido de Pasante
* Generar Convenio Particular
* Ver Alumno
* Ver Pedido Pasantía
* Ver Empresa

**Procedimiento de Técnica de estimación Planning Poker**

* Todo el equipo ágil participa en la sesión. Cada integrante recibe un manojo de cartas de Planning Poker cada carta tiene un valor de la secuencia de números de Fibonacci.
* El Product Owner presenta el elemento del Product Backlog que se van a estimar.
* El equipo de desarrollo discute el elemento a estimar, en primera instancia se elige el elemento "canónico" que lleve menos esfuerzo, capacidad e incertidumbre para ello se realiza preguntas PO. Este último responde a las preguntas.
* Cada estimador en privado selecciona una carta que representará su estimado.
* Cada estimador muestra su carta.
* Si todos han seleccionado la misma carta, se habrá logrado el consenso. El resultado se le asigna como estimación al elemento de Product Backlog.
* Si los estimados no fueron los mismos, se realiza una discusión para comprender las premisas tomadas en cada estimación o malos entendidos. Por lo general se comienza por preguntar al estimador más alto y al más bajo que expliquen y justifiquen sus estimados.
* Luego de la discusión se regresa al paso 3 para que estimar los demás elementos. usando como comparación la canónica. El ciclo se repetirá hasta lograr el consenso.