

Semana Uno

Fecha	20 - 24 febrero 2023
Nombre	Olivia Yuyu Maceda Pérez

Ejercicios Xideral

- Explica e implementa un programa utilizando polimorfismo.

Ofrecer la herramienta adecuada a las personas con alguna discapacidad, de acuerdo con sus necesidades se le asignará una herramienta especial para una mejor interacción en el lugar que ofrezca algún servicio.

Con ayuda del polimorfismo, tomando en cuenta que un método puede tomar muchas formas, en la programación orientada a objetos y con la clase discapacidad, es fundamental que la discapacidad tenga por lo menos una herramienta para un mejor desenvolvimiento. Por ello, asignamos una herramienta dependiendo el tipo de discapacidad con la que cuente, es decir, nuestro método se acopla a la discapacidad para asignar la herramienta.

LINK

- Del ejercicio anterior, impleméntalo con clases abstractas e interfaces.

Clase abstracta: LINK

Interfaces: LINK

- Explicar todos los pasos de Scrum.

Una vez mostrado lo que se logró en el sprint anterior, se echa un vistazo atrás para cuestionarse qué marchó bien, qué pudo mejorar y qué se puede mejorar. **Juzgar el proceso** para encontrar posibles mejoras en siguientes proyectos.



Al final de la reunión, equipo y Scrum Master deben acordar una mejora al proceso que implementarán en el siguiente sprint. Esa mejora al proceso, también llamada **Kaizen**, debe incorporarse en la bitácora del sprint siguiente, con pruebas de aceptación.

tomando en cuenta la experiencia del equipo con los impedimentos y mejoras del proceso

Retrospectiva del sprint

10



Comenzar de inmediato el ciclo del siguiente sprint

Revisión o demostración del sprint

En una reunión, donde todos pueden asistir (además de las partes involucradas en el proyecto) el equipo muestra lo que realizó durante el Sprint, satisfaciendo la definición de "Terminado"; puede no ser un producto terminado, pero si una función concluida de uno de ellos.



Scrum diario



Cada día, a la misma hora, durante no más de 15 minutos* (obligatorio); equipo y Scrum Master contestan:

- ¿Qué hiciste ayer para ayudar al equipo a terminar el sprint?
- ¿Qué harás hoy para ayudar al equipo a terminar el sprint?
- ¿Algún obstáculo te impide o impide al equipo cumplir la meta del sprint?

Con el fin de ayudar al equipo para saber dónde se encuentran y colaborar si es necesario. Pero, tomando en cuenta que las tareas no se asignan desde arriba, el equipo es autónomo.

Vuelve visible el trabajo



Llevar un registro de las tareas, un ejemplo tabla Scrum (pendientes, en proceso, terminadas) avanzando por la tabla conforme se van concluyendo las tareas. Otro ejemplo, diagrama de finalización.

Planeación del sprint

*Primera de las reuniones de Scrum.

Responsable del producto, equipo y ScrumMaster se reúnen para planear el sprint. (Extensión fija, inferior a un mes) examinando el inicio de la bitácora y pronostica cuánto tiempo puede llevarse en este sprint.



Afinar y estimar la bitácora del producto

Los encargados de la bitácora deberán estimar cuánto esfuerzo implicará cada tarea. El equipo debe de revisar a detalle para verificar que sea viable la propuesta. Además, deberán cuestionarse para que la tarea sea mostrada, demostrada y (lo ideal) entregarse.



5

Elegir un responsable del producto



Elige a una persona que tenga la habilidad de ser visionaria respecto a tu crecimiento, tomando en cuenta varios factores realistas para lograr el objetivo.

2 Seleccionar equipo



Elige al equipo de acuerdo con las habilidades necesarias del proyecto para colaborar con el responsable del producto para transformar esas ideas en hechos.
*Conformas de 3 a 9 personas.

3 Elegir Scrum



Elegir a una persona para preparar al equipo con un enfoque Scrum, apoyándolo a priorizar las tareas y desechar lo que no es importante.

4 Crear y priorizar una bitácora del producto



Enlistar todas las tareas de acuerdo a las prioridades para hacer realidad la visión del producto. La lista es única y la puede modificar el responsable del producto durante el período de vida del producto representando lo que se puede hacer y lo que la gente necesita.

- Explica y programa un singleton.

Un singleton es un patrón de diseño utilizado para crear únicamente un objeto y que se creen objetos innecesariamente.

En el ejemplo, supongamos el escenario de una oficina de correos postales. Donde varios usuarios (dependiendo su domicilio - colonia) tendrá asignado una oficina de correos postales. Es decir, no necesitamos una oficina de correos para cada domicilio, sino seccionadas por región es como se asigna la oficina de correos.

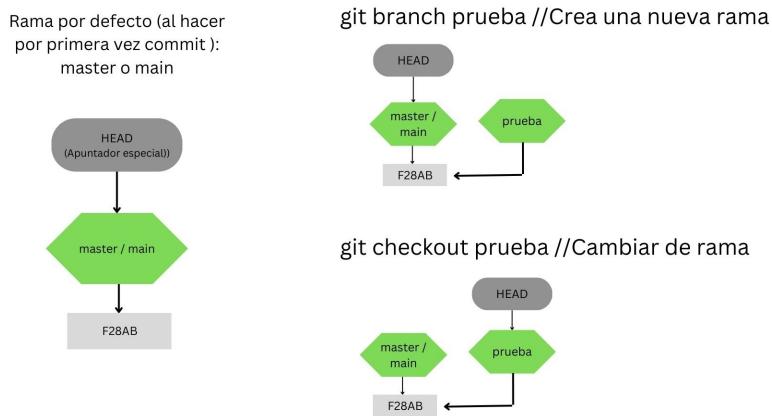
LINK

- Investiga y explica qué es una branch, merge y conflicts en git.

Branch: (Rama)Es un apuntador móvil que apunta a uno de los commits hechos. La rama por defecto de Git es la rama master. Es decir, es una 'nueva' o separada versión de la rama main/master.

Las ramas te permite desarrollar características, arreglar bugs o experimentar de manera segura sin necesariamente impactar la rama main.

Siempre se creara una rama de una rama existente.



Merge: Fusionar uno o más ramas dentro de la rama que tienes activa, uniendo los cambios hechos en tu rama con las otras.

Conflicts: Cuando dos archivos tienen cambios que son tan similares que git no sabe exactamente como resolverlo

- Programa un inyector de dependencias (Con constructor, setter y getter)

Fuentes: