

Resumen SCRUM

Para poder comenzar a usar Scrum, como primer paso es necesario elegir un responsable del proyecto/producto, este será el encargado del que hacer, producir o lograr, teniendo ya un responsable se llevará a cabo de la selección del equipo, los integrantes deberán contar con todas las habilidades necesarias para lograr los objetivos, estos equipos deben ser formados entre 3 y 9 personas, grupos pequeños es lo recomendable.

Ahora viene la selección del scrum master, esta persona será la encargada de capacitar al resto del equipo en el enfoque Scrum y que ayudará al equipo a eliminar todo lo que lo atrasa.

Para poder llevar un mejor control es necesario crear una bitácora del producto, que es una lista de todo lo que debe hacerse para lograr los objetivos, la bitácora evoluciona durante el periodo de vida del producto, esta bitácora tiene todo lo que el equipo debe hacer y el orden de prioridad de las actividades, El equipo debe examinar cada elemento de la bitácora, para determinar si es viable, cada elemento mostrado en esta debe poder mostrarse, demostrarse y entregarse, se recomienda que no se calcule en tiempo(horas), si no en tamaño relativo (pequeño, mediano, grande), otra forma recomendable de calcular es mediante la serie de Fibonacci.

La planeación de sprint, esta es la primera de las reuniones de SCRUM, todos los involucrados en el proyecto(líder, equipo y scrum master) deben estar presentes, Los sprints son siempre de extensión fija, inferior a un mes, utilizando de base la bitácora se pronostica cuando llevar a cabo estos sprints. Si el equipo ha pasado por varios sprints, debe considerar el número de puntos que acumuló en el más reciente. Este número se conoce como velocidad del equipo. El Scrum Master y el equipo deben tratar de aumentar ese número en cada sprint. Una vez que el equipo se compromete con lo que cree que puede terminar en un sprint, no puede cambiar ni crecer.

Se recomienda crear una tabla de scrum con tres columnas: Pendiente, En proceso y Terminado, para que el equipo vaya avanzando conforme la tabla, otra forma de volver visible el trabajo es crear un diagrama de finalización, En un eje aparece el número de puntos que el equipo introdujo en el sprint y en el otro el número de días. Cada día, el Scrum Master suma el número de puntos completados y los grafica en el diagrama de finalización

Otro punto de scrum es que Cada día, a la misma hora, durante no más de quince minutos, el equipo y el Scrum Master se reúnen y contestan tres preguntas:

- ¿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?

Una de las funciones del scrum master es eliminar los obstáculos, o impedimentos, contra el progreso del equipo.

Existe otra reunión donde el equipo muestra lo que hizo durante el sprint, esta puede ser una reunión abierta a todos los interesados en el proyecto, donde se hace una demostración de todo lo que se pudo llevar a cabo durante el sprint,

el equipo debe mostrar únicamente lo que satisface la definición de Terminado, Esto puede no ser un producto terminado, pero sí una función concluida de uno de ellos.

Después se realiza una retrospectiva del sprint, para analizar que se puede mejorar, Es crucial que la gente, como equipo, asuma la responsabilidad de su proceso y de sus resultados y busque soluciones también como equipo. El equipo debe tener fortaleza para tocar los temas que le incomodan de un modo orientado a la solución. Este debe tener la madurez de oír la realimentación, aceptarla y buscar una solución.

Al final de la reunión, el equipo y el Scrum Master deben acordar una mejora al proceso que implementarán en el siguiente sprint. Esa mejora al proceso se le denomina kaizen, después de esto comienza de inmediato el ciclo del siguiente sprint, tomando en cuenta la experiencia del equipo con los impedimentos y mejoras del proceso.

Excepciones

Excepciones JAVA

En java existe una clase llamada Throwable que es una clase la cual contiene todos los problemas/errores que puede llegar a tener java, este a su vez se divide en 2:

- Error
- Excepción

Los errores son subclases de Throwable que son problemas o errores que no se pueden recuperar o problemas que no se pueden resolver, son aquellos problemas graves que la aplicación no puede resolver, se podrían denominar errores externos, por ejemplo, disco dañado, caída de internet, memoria agotada, etc.

Las excepciones también son subclases de Throwable, son aquellos errores o problemas que hacen que el programa no compile de una forma correcta o se rompa, este se puede dividir en 2:

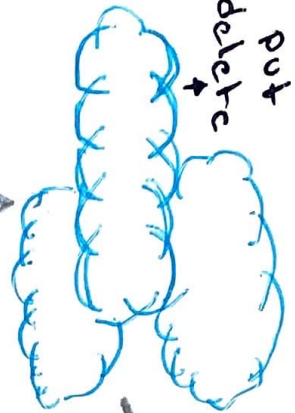
- Runtime Excepcions
- Other Excepcion

Las runtime excepciones también conocidas como Unchecked son aquellas que el compilador nos obliga a manejarlas/controlarlas, no se nos obliga a darles un tratamiento, normalmente son tratadas mediante un try & catch

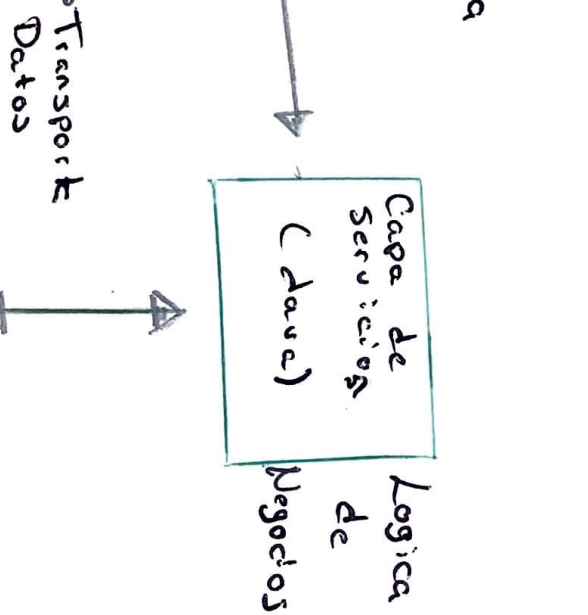
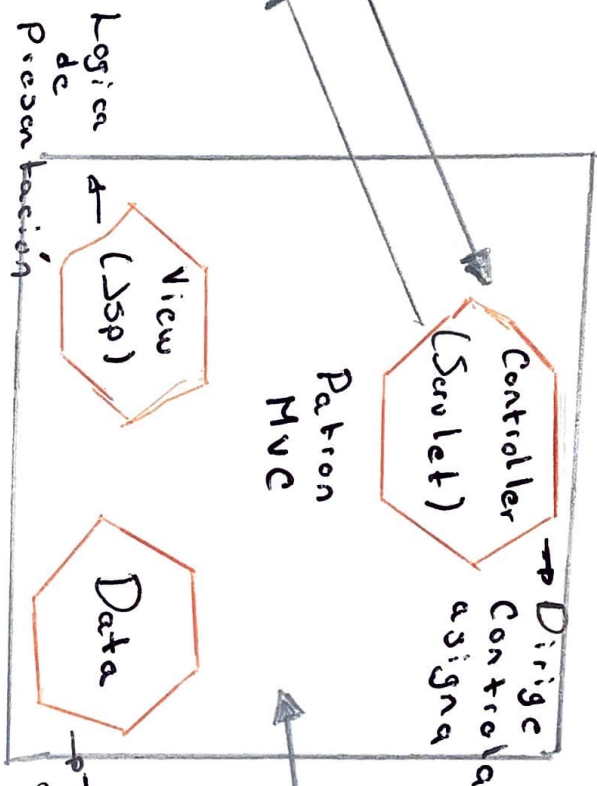
Las Other excepciones también conocidas como Checked, son aquellas que el compilador nos obliga a manejarlas/controlarlas, estas también pueden ser tratadas mediante un try & catch, aunque también se le puede dar la responsabilidad de ella al throw que es su padre.

Http 1 Http 2

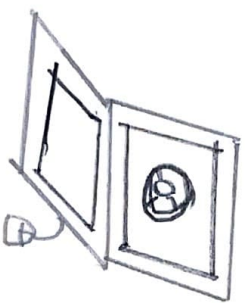
get
post
put
delete



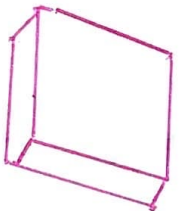
Request
Response
Html
Text
Json
xml
etc.



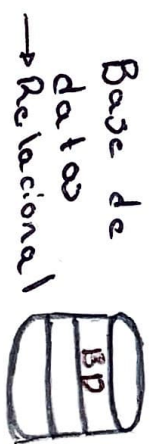
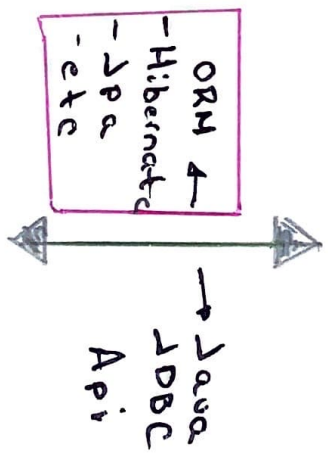
Transport Datos



→ HTML
Java Script
CSS



Servidor de Aplicaciones
→ Tomcat 9
Java SDK



Base de datos Relacional
→ Oracle
→ MySQL