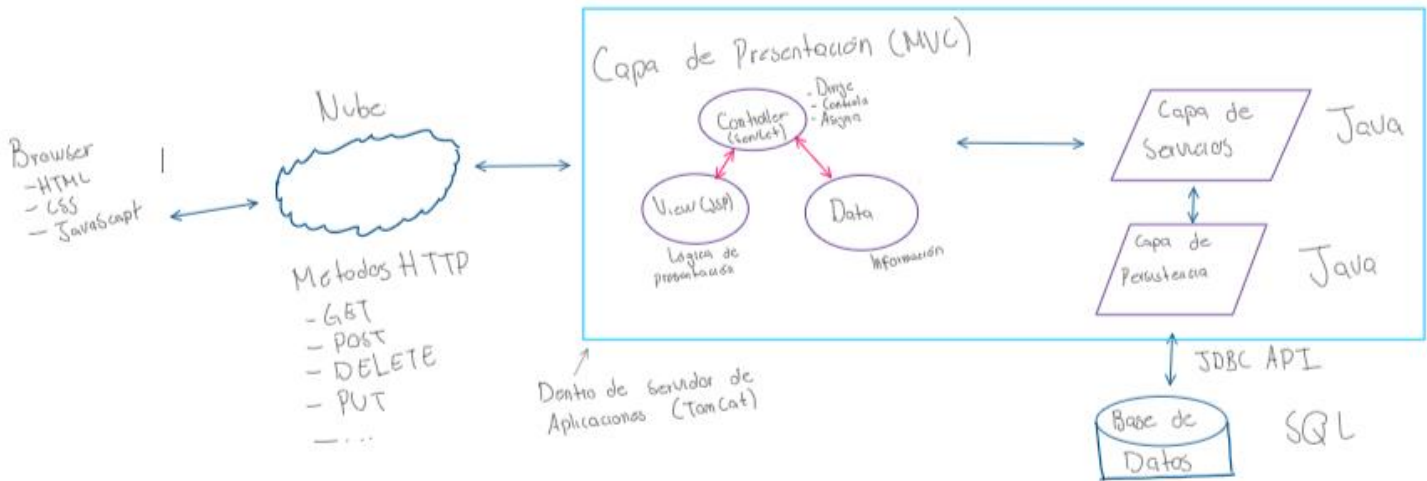


Examen 2 – Armando Molina Rincón

1. Dibujar un diagrama de la arquitectura de una aplicación web



2. Realizar un programa utilizando el patrón Composite y pruebas unitarias.

<https://github.com/Amolin10/BecaXideralExamenes>

3. Explica la forma de implementación de la metodología Scrum.

Existen 11 pasos para implementar la Metodología Ágil SCRUM.

Paso 1: Seleccionar una persona responsable del producto.

El responsable del proyecto posee la visión de lo que se va a hacer, producir o lograr. Toma en cuenta los riesgos y las recompensas.

Paso 2: Seleccionar un equipo.

Los integrantes del equipo deben contar con las habilidades necesarias para tomar la visión y hacerla realidad. Los equipos son pequeños, de tres a nueve personas.

Paso 3: Elige un **Scrum Master**.

Esta es la persona que capacitará al resto del equipo en el enfoque Scrum.

Paso 4: Bitácora del producto.

Es la guía para completar la visión. Contiene todo lo que se debe hacer en orden de prioridad. El responsable del equipo debe tomar decisiones sobre la prioridad de las actividades.

Paso 5: Afinar y estimar la bitácora del producto.

El equipo debe examinar cada elemento de la bitácora y determinar si es viable o no.
El responsable del producto asigna puntos a cada actividad.

Paso 6: Planeación del **Sprint**.

El equipo, el Scrum master y el responsable del producto planean el sprint, los sprints son de duración fija, generalmente se realizan uno o dos sprints semanales.

La velocidad del sprint se determina por el número de puntos alcanzado en el sprint anterior.

Una vez que el equipo se compromete con lo que cree que puede terminar en un sprint, esto no puede cambiar ni crecer.

Paso 7: Volver visible el trabajo.

Se puede utilizar una tabla de Scrum o un diagrama de finalización. En la tabla de scrum, hay tres columnas: Pendiente, En Proceso y Terminado. En el diagrama de finalización se grafican los puntos vs los días. Cualquiera de los dos es actualizado diariamente por el scrum master.

Paso 8: Scrum diario.

Es una reunión diaria de no más de 15 minutos, en ella se responden las siguientes 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?

El scrum master se encarga de eliminar los obstáculos.

Paso 9: Revisión del sprint.

En esta reunión el equipo muestra el trabajo que realizó durante el sprint.

Paso 10: Retrospectiva del sprint.

Pensar en que se ejecutó bien y que puede mejorarse en el proceso.

Paso 11: Siguiendo ciclo.

Comenzar de inmediato el próximo sprint.

4. Explica que son las excepciones.

Las excepciones permiten mantener el control del programa y ejecutar líneas de código alternas en caso de que ocurra una excepción.

Las excepciones son los problemas que pueden ocurrir en tiempo de ejecución o tiempo de compilación.

Excepciones Checked

Ocurren en tiempo de compilación, por lo que el compilador obliga a tratarlas. Esto puede hacerse encerrando la línea que lanza la Excepción en un bloque `try-catch` o bien, lanzando la Excepción en el método que la contiene, obligando al método que invoca a este, tratar la excepción. Las excepciones Checked heredan de *Exception* de *java.lang*.

Excepciones Un Checked

Ocurren en tiempo de ejecución, heredan de *RuntimeException* de *java.lang*

El compilador no obliga a tratarlas, sin embargo, en caso de que se lancen, se perderá el control del programa y se detendrá. De igual manera, pueden tratarse con un bloque *try-catch*, o bien, lanzando la excepción al método que la contiene.

El bloque *try-catch*

Es una sentencia de Java para tratar las excepciones.

- Dentro del `try` se colocan las líneas que pueden provocar la excepción
- Dentro del `catch` las líneas de código que se ejecutan si se lanza la excepción
- Y un bloque opcional `finally` que se ejecuta ocurra o no la excepción.

La sentencia `try with resources` cierra automáticamente los recursos utilizados en el bloque `try` al finalizar la ejecución de todas las instrucciones dentro del bloque, por lo que no debemos preocuparnos por escribir un bloque *finally* donde intentamos cerrar las conexiones a los recursos, como se hacía en versiones anteriores a Java 7.

A partir de Java 7, también, existe `multicatch`, que permite acumular varios tipos de excepciones en el mismo bloque *catch*.

La sentencia *throws* sirve para lanzar la excepción al método superior que invoque al que la contiene, obligándolo a tratarla o lanzarla.

5. Realiza un programa que se conecte a una base de datos utilizando Maven, JSP y el patrón MVC. El programa debe permitir consultas, actualizaciones, altas y bajas de datos.

<https://github.com/Amolin10/BecaXideralExamenes>

El script de la base de datos se encuentra dentro del proyecto Examen2Maven/sql/deportista.sql