**Actividad de aprendizaje**

**Descripción de la actividad**

En equipos de 4, los cuales se organizarán en orden alfabético, desarrolle una aplicación web en base a requerimientos reales. Entreviste a una persona (o varias) que tenga necesidad de una aplicación web, por ejemplo, en una tienda, en un restaurant, en la universidad, en su casa, o algún amigo.

Si el número de estudiantes total del curso no es múltiplo de 4, el último equipo se conformará de la siguiente manera: si sobran 3 alumnos, se conformará un nuevo equipo, caso contrario lo últimos dos o un estudiante se unirán al último equipo. Los números de equipo se asignan automáticamente en orden alfabético, los primeros cuatro estudiantes son el Equipo1 (T1), y así sucesivamente.

Durante la entrevista, tome apuntes de los requerimientos del usuario a manera de historias de usuario, pídale que le cuente como realiza sus actividades en la actualidad sin el uso de la computadora, una vez definidas las historias de usuario, transfórmelas en un conjunto de ítems/features del sistema para colocarlos en el product backlog (al menos debe haber 4 items 1 POR CABEZA), a estos features se los conocerá como requerimientos funcionales del sistema.

1 PERSONA O EN GRUPO

En equipo, realice un diseño rápido usando diagramas UML, un diagrama de casos de uso para modelar la funcionalidad y los usuarios, un diagrama de clases para modelar la estructura de los datos del sistema, y la arquitectura del sistema.

EN GRUPO

Deben crear una base de datos en MongoDB Atlas para que sea compartida por los miembros de su equipo. A continuación, cada miembro del equipo, implemente su cliente (HTML, JavaScript y CSS) y su servidor web (JSP + MongoDB) correspondiente (al menos debe haber 1 feature por cada desarrollador TOTAL 4), tome en cuenta que el CSS ocupado, debe ser el mismo para los cuatro desarrolladores, para que las páginas del proyecto se miren similares.

GRUPO

NOTA: Las operaciones CRUD, si bien son necesarias en la aplicación a desarrollar, por si solas no representan un requerimiento funcional per se, por lo tanto deben ir acompañadas de alguna regla de negocio, el cálculo de algún valor, por ejemplo: costo total de una factura, cantidad y selección del tipo de alimento para un animal dependiendo de su peso y edad, el salario de un empleado dependiendo de sus horas trabajadas, etc.

Una vez desarrolladas las funcionalidades, al menos una por cada integrante, en quipo deben integrar toda su aplicación para colocarla en un solo servidor Web, ya sea usando Payara, Apache Tomcat, u otro.

EN PAYARA POR DIOSITO

La ejecución del programa se grabará en un video en YouTube por parte de todos los integrantes del equipo, se describe la funcionalidad mientras se ejecuta la aplicación Web desarrollada. Al inicio del video se debe mencionar los nombres de los integrantes y el tema desarrollado.

**GRUPO EL VIDEO**

**Formato de entrega de los archivos**

* Un google doc, creado por el líder del equipo, el cual contiene el número y nombre del equipo, capturas de pantalla de cada elemento de los criterios de evaluación. En este documento al inicio, comparta el enlace al repositorio GitHub y al video YouTube de la actividad. Comparta el documento con el docente de la asignatura.

(ESTO DEBE DE IR EN WORD “DEFINICIÓN DE PROYECTO”)

* Un repositorio en GitHub. El código será subido a un repositorio público (GitHub) creado por el líder de cada equipo. Se debe compartir el enlace al código en el documento de entrega. Los miembros del equipo y el docente de la asignatura serán invitados como colaboradores del repositorio.
* La estructura del repositorio es la siguiente:
  + 01-Definition:
    - Documento de la definición de su proyecto
  + 02-Requirements :
    - SRS (IEEE-830) o lista de historias de usuario o features del sistema y otro documentos que cree pertinente respecto al levantamiento de requerimientos de las necesidades del usuario AQUI PONER ENTREVISTA
  + 03-Documentation:
    - Cualquier documento necesario que no sea diseño o requerimientos del sistema.

AQUI PONER CAPTURAS DE LAS PANTALLAS, RETICULAS, LOS DISEÑOS DCU, NAVEGACIÓN, REPORTE DE FALLOS (OPCIONAL), TODO ESTO SE PUEDE HACER COLABORATIVAMENTE EN <https://my.moqups.com/>

* + 04-UMLDiagrams
    - 01-UseCases: archivos vpp (Visual Paradigm) y archivos pdf
    - 02-ClassDiagram: archivos vpp y archivos pdf
  + 05-Tests:
    - Documentos de pruebas del sistema, de no contar con pruebas automatizadas (WORD “PRUEBAS DE SISTEMA”)
  + 06-Code:
    - Código del proyecto Netbeans, scripts SQL
  + 07-Other:
    - Cualquier otro documento adicional. Incluyendo su actividad de evaluación en formato Word a partir del google doc generado. (WORD “ACTIVIDAD DE EVALUACION”, VIDEO YOUTUBE)

**Graphical user interface, text, application, email

Description automatically generated**

**Nombre del repositorio**

http://github.com/usuario/Team1apellidonombreTeamLider

**Ejemplo**: http://github.com/elascano/Team1lascanojorge

**Criterios de evaluación**

1. Repositorio Github (Estructura y colaboración individual) 1/1
2. Idea (entrevista) y Lista de features (product Backlog) 1/1
3. Diagramas (Clases, Casos de Uso, Arquitectura, UML) 1/1
4. Base de datos en MongoDB Atlas (en la nube) cagado/1
5. Diseño del cliente y cuatro reglas de negocio /1
6. Ejecución de la aplicación 1/1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NOTA TOTAL: /6

**Extensión máxima**

No aplica