

SEGUNDO PARCIAL DE PROGRAMACIÓN WEB

tiendau servicio
id : int(11)
nombre : varchar(200)
descripcion : text
tienda : int(11)

tiendau cliente
id : int(11)
nombre : varchar(100)
email : varchar(100)
clave : varchar(100)

tiendau tienda
id : int(11)
nombre : varchar(100)
lema : varchar(200)
descripcion : text
email : varchar(100)
clave : varchar(100)
propietario : varchar(100)
facebook : varchar(100)
web : varchar(100)
imagen : varchar(500)

tiendau seguir
cliente : int(11)
tienda : int(11)

El backup está en SQL en la una base de datos MySQL. Se debe dejar el usuario **root** y sin clave al momento de generar el WAR para poder desplegarlo en mi servidor.

Leer penalidades al final del documento.

La Universidad con el ánimo de apoyar a su comunidad desea habilitar un espacio para promocionar los emprendimientos de sus estudiantes, los cuales ofrecen servicios varios como domicilios, comidas y demás. Para poder llevar a cabo este requerimiento se ha desarrollado un modelo de datos como el que se observa en la siguiente imagen:

En base al modelo anterior se deben desarrollar las siguientes actividades:

- (Valor 1.0) Realizar la conexión a la base de datos para todas las entidades propuestas en el modelo y realizar un archivo de prueba (-0.5) que garantice el funcionamiento del modelo. El archivo de prueba debe realizar:
 - Creación de un cliente
 - Creación de una tienda
 - Creación de un servicio
 - Seguir una tienda por un cliente
 - Actualizar una tienda
- (Valor 0.8) Implementar la interfaz que permite mostrar el listado de tiendas disponibles con los servicios de nuestros estudiantes. Ojo asociar los datos de la base de datos al formulario. Los enlaces deben enviar a



Facebook si está registrado, a la página web y a los servicios de la tienda (-0.2).

- (Valor 1.0) Implementar el formulario que permita el registro de una tienda con el formulario que se muestra a continuación:



Registro de Tienda

Nombre

Lema

Descripción

Email
crangarita

Password
....

Propietario

Este formulario debe almacenar la información en la base de datos y realizar algunas validaciones.

- (-0.2) El email debe ser único, por lo tanto, se debe validar al momento de registrar el usuario se debe verificar que no exista. En caso de encontrarse registrado se debe retornar un mensaje que indique que el email se encuentra registrado.
- (-0.2) Se debe validar que la imagen realmente exista en la dirección ingresada.
- (0.2) Después de registrado se debe enviar un email con un mensaje indicando que la tienda fue registrada correctamente.

- (Valor 0.8) Implementar la interfaz de login. Este sistema debe validar:
 - (-0.2) Si el usuario es **tienda** debe redireccionarlo a una interfaz nueva donde se observan los servicios de la tienda.
 - (-0.2) Si el usuario es **cliente** redirecciona a la interfaz principal de servicios.



Bienvenido

Email
crangarita

Password
....


☐ Recordar Password

INGRESAR

SERVICIOS DE \${NOMBREDETIENDA}

700 x 400	700 x 400	700 x 400	700 x 400
Servicio Uno Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam viverra euismod odio, gravida pellentesque urna	Servicio Dos Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam viverra euismod odio, gravida pellentesque urna	Servicio Tres Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quos quisquam, error quod sed cumque, odio distinctio	Servicio Cuatro Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam viverra euismod odio, gravida pellentesque urna

- (Valor 0.6) Cargar listado de servicios de la tienda en base a la siguiente interfaz. En caso de no haber servicios mostrar un mensaje indicando que no hay servicios (-0.2). La interfaz anterior debe ser cargada desde el punto 4a

al momento de ingresar una tienda (-0.2). Adicional debe ser cargada desde la interfaz de tiendas, al hacer clic en el botón  (-0.2).

- (Valor 0.6) Hacer un formulario para el registro de un servicio perteneciente a una tienda que este logueada.

PENALIDADES

No usar JSTL (-0.5)

No usar MAVEN (-1.0)

No usar Github (-1.0)

Se deben hacer 5 commit, el de inicio y uno cada media hora. Por cada commit que no hagan -0.1. Igual deben empaquetar un archivo WAR y enviarlo a través de Gmail (-0.5).