

Programación N-Capas - Taller I

Sistema de inicio de sesión
29 de marzo de 2023

Contexto:

Una empresa de gestión de talento humano los ha buscado como equipo para mejorar su sistema informático con el que administran a sus empleados.

Durante una auditoría a los módulos de todo el sistema, la empresa se percató que cualquier persona podía acceder a todas las herramientas y funciones que el sistema ofrece; incluidas las características relacionadas con procesos administrativos, de contratación, planilla, entre otros.

Como primera fase del proyecto de reestructuración y mejoramiento, les han solicitado que implementen el módulo de autenticación, que asegure un alto nivel de seguridad.

Se requiere saber de los empleados la siguiente información: código de empleado (@@##### -> AD123400), nombre, apellido, fecha de contratación, si está activo o no y su rol (Administrador o normal). Como parte del módulo se solicita almacenar una contraseña con el fin de facilitar la gestión de los procesos.

La misma empresa, durante las reuniones, planteó varios escenarios posibles a considerar en el inicio de sesión de sus sistema:

- El usuario debe proporcionar su código de empleado y la contraseña asociada a él. (2 pts.)
- Un usuario inactivo no puede iniciar sesión dentro de la aplicación (1 pts.).
- Por políticas de la empresa, un empleado es registrado en el sistema 1 mes antes de comenzar a trabajar. La fecha de contratación se establece en su primer día laboral. Y se ha solicitado que estas personas no puedan iniciar sesión hasta que se cumpla esta fecha (2 pts.).
- El sistema tiene dos casos de uso: El de usuario normal (2 pts.), y el de administrador (2 pts.). El inicio de sesión debe de distinguir entre roles y retornar las vistas necesarias para cada rol específico.

Se ha solicitado que, cuando un usuario ingrese mal las credenciales, o por los motivos anteriores no tenga acceso al sistema, se debe de retornar una vista con el código de error "404 no encontrado" (1 pts.).

Definición:

Haciendo uso de los conocimientos adquiridos en la materia, diseñe una solución a la problemática planteada anteriormente, y tomando en cuenta los siguientes aspectos:

- Debe de ser una aplicación web, desarrollada con el framework Spring Boot, y con el lenguaje de programación Java. (Si desea utilizar Kotlin, debe de solicitar su aprobación)
- Como motor de vistas se debe utilizar la librería Mustache.
- En todo momento se debe de llevar un control de versiones, utilizando git como herramienta.
- Todas las clases deben de estar organizadas en paquetes con peso semántico.
- Se deben realizar validaciones tanto a nivel de cliente como de servidor, tanto formato como lógica
Por ejemplo: campos vacíos en el cliente o validez de la contraseña en el servidor.
Nota: En el servidor solo realizar validaciones de lógica

Aclaraciones del contexto del problema:

- Los datos de los empleados deben de almacenarse en una colección de datos.
- El módulo de usuario normal simplemente mostrará el nombre del empleado y la hora del servidor. El módulo de administrador, además de hacer lo que hace un usuario normal, mostrará la lista de empleados registrados, junto a la fecha de contratación y el rol que posee. Ambos módulos deben de estar sobre la misma ruta.
- Los maquetados deben priorizar la definición de las interacciones con el usuario.

Metodología de entrega

- Código fuente: será entregado haciendo uso de la plataforma github classroom, donde inscribirán el grupo de acuerdo al correlativo que seleccionaron previamente en moodle. El nombre de su grupo en classroom deberá corresponder a la siguiente plantilla: Grupo de trabajo <# de grupo>.
Para dar por recibido un trabajo se deberá de realizar un release en el repositorio, indicando las características agregadas, y las rutas implementadas.

La fecha de entrega será el día 29 de marzo de 2023 a las 20:10. Además deberá, un integrante del grupo, exponer el entregable al docente (será la única forma de obtener la nota)

Fecha máxima para presentar: Viernes 31 de marzo de 2023, a las 12:00

Requisitos de entrega

El objetivo principal de la evaluación es aplicar los conocimientos de las clases para desarrollar el módulo de autenticación del problema planteado. Para ello, y garantizar la correcta aplicación de estos; se requerirán los siguientes criterios para tomar como válida la solución

Área de evaluación	Criterio	Descripción
Formato y entrega	Uso de Github Classroom	Configura correctamente el nombre del grupo acorde a las indicaciones presentadas, y realiza el release de su entrega indicando la información que se solicitó.
	Uso de commits	Todos los mensajes cumplen con un estándar y agregan valor a la entrega. p.e. "Add: basic configuration". Añaden al menos 5 commits
Servicio Web	Manejo, proceso y respuesta de peticiones	Se definen los controladores necesarios para solucionar la problemática planteada, estableciendo por cada uno los parámetros de entrada, un proceso claro de la información, y la salida o respuesta del mismo.
	Uso de objetos para la transferencia de datos	Se utiliza el patrón de diseño DTO para transferir los datos en cada interacción que exista entre los componentes de la aplicación.
	Server Side Rendering	Los controladores, luego de procesar la información generan las vistas acordes a los resultados calculados, y garantizan que los datos inyectados optimicen su propio consumo y no expongan la seguridad de la aplicación.
	Uso adecuado de POJOS	Se manejan uno o varios modelos que poseen todos los atributos que se solicitan en la descripción, eligiendo los tipos de datos indicados para cada uno.
	Validaciones Server Side	Se realizan todas las validaciones indicadas en el contexto del problema, cambiando el flujo de interacción dependiendo de cada caso de aplicación.
Vistas	Maquetado y visualización	Todas las vistas cumplen con lo requerido en el contexto, y garantizan una buena experiencia de usuario.
	Validaciones Client Side	Se realizan validaciones necesarias para garantizar que en los procesos a realizar dentro los controladores no existan errores no controlados. P.e. campos vacíos.
Código fuente	Orden, identificadores semánticos e indentado	Todo el código entregado presenta un orden lógico, se utilizan identificadores semánticos para las variables / métodos, y está correctamente indentado facilitando la lectura del mismo.
	Uso de paquetes	Las clases están agrupadas por paquetes con peso semántico, permitiendo separar dentro del mismo proyecto las capas y módulos de la aplicación.
	Tecnologías utilizadas	Hace uso del Framework Spring para la base de la aplicación web, y mustache como motor de vistas en el proceso SSR.