



Instituto Tecnológico de Chetumal



Alumno:

Benito Tecuautzin Belen Del Rosario
Edwin Ernesto Canul Gongora

No.Ctrl:

19390239
19390062

Grupo:
I8U

Carrera:
Ing. Sistemas Computacionales

Asignatura:
Framework para Back-End

Maestro:
May Canché Isaías

Práctica1 Spring MVC: Para esta práctica se hizo un proyecto web con spring boot, en el que se utilizó la versión de java 11, packaking jar y el type Maven. Para las dependencias se utilizaron Spring Boot DevTools, Thymeleaf y Spring Web. Para el desarrollo del proyecto se utilizó anotaciones (@) y creó paquetes: Controller para todos los endpoints que tenga nuestra aplicación y la vista de esta, en web. @ModelAndView, vModelMapy y @Model para pasar los datos a la vista por un html. @RequestMapping para procesar solicitudes HTTP con patrones de URL especificados. Models para las representaciones de nuestro modelo de datos (entidades). La directiva if de Thymeleaf y each para utilizar condiciones en los datos. Además de @ModelAttribute, @RequestParam, @PathVariable, @Value y el uso de recursos estáticos y estilos CSS.

Práctica2 Inyección de dependencias: Para esta práctica se hizo un proyecto web con spring boot., con el fin de hacer un ejemplo de creación de Facturas, en que se utilizó la versión de java 11, packaking jar y el type Maven. Para las dependencias se utilizaron Spring Boot DevTools, Thymeleaf y Spring Web. Para el desarrollo del proyecto se utilizó anotaciones (@) y creó paquetes: Controller para todos los endpoints que tenga nuestra aplicación y la vista de esta, en web. Model para las representaciones de nuestro modelo de datos (entidades). @Autowired que importa automáticamente objetos a la clase, y la clase inyectada también es administrada por el contenedor Spring. Además del uso de @Primary, @Qualifier, @Bean, @SessionScope, y la inyección de dependencias vía constructor y métodos setters y la inyección mediante la interface.

Práctica3 Formularios Thmeleaf y Data Binding: Para esta práctica se hizo un proyecto web con spring boot., con el fin de hacer un formulario de registro para usuarios con estilos css Bootstrap. Para el desarrollo del proyecto se utilizó la versión de java 11, packaking jar y el type Maven. Para las dependencias se utilizaron Spring Boot DevTools, Thymeleaf, Spring Web y Validation. Las anotaciones (@) y paquetes usados fueron: Controller para todos los endpoints que tenga nuestra aplicación y la vista de esta, en web. Model para las representaciones de nuestro modelo de datos (entidades). Service para agregar ahí las clases que respondan a la funcionalidad y lógica. Uso de @SessionAttributes para mantener los datos durante los ciclos de form, @Size para rangos, @Email para validar correo electrónico,

@Pattern para expresiones regulares, @Max, @Min y NotNull para validación de números enteros. @Past, @Future para validación de fechas con input. @InitBinder para formateo de fechas. Por último, listado de select con Map, implementación de checkboxes para tipo de países, roles, valores booleano e implementación de mensajes de errores personalizados y convertidor de mayúsculas.

Práctica4 Interceptores HTTP: Para esta práctica se utilizó el proyecto anterior y se implementó con interceptores http de tiempo transcurrido, fragment para reutilizar código de thymeleaf, interceptor de peticiones post http return en post handler y retornando false en un receptor. Con importaciones HttpServletRequest, ModelAndView, Random.

Práctica5 Interceptores: Horario de acceso a clientes: Para esta práctica se hizo un proyecto web con spring boot., con el fin de hacer un index html que visualizara el encabezado de si esta fuera del horario de atención al cliente o no, junto con las horas disponibles de atención. Para el desarrollo del proyecto se utilizó la versión de java 11, packaging jar y el type Maven. Para las dependencias se utilizaron Spring Boot DevTools, Thymeleaf y Spring Web. Las anotaciones (@), paquetes e importaciones usados fueron: Model para las representaciones de nuestro modelo de datos (entidades). el método handler, interceptor, NullPointerException, HttpServletRequest, ModelAndView, Calendar, HttpServletResponse, HandlerInterceptor, @Value, @Override, etc.

Práctica6 Manejo de errores: Para esta práctica se hizo un proyecto web con spring boot., con el fin de crear un html de error 404 y 500 personalizados. Para el desarrollo del proyecto se utilizó la versión de java 11, packaging jar y el type Maven. Para las dependencias se usaron Spring Boot DevTools, Thymeleaf y Spring Web. Utilizamos la anotación @ControllerAdvice para manejo de excepciones., personalizamos la página de error para la excepción NumberFormatException. Creamos las clases model y service para personalizar el manejo de excepción y añadimos el método handler para ver la información.

Práctica7 Spring MVC & Data JPA: Para esta práctica se hizo un proyecto web con spring boot, con el fin de crear un formulario de registro, un listado de clientes con funciones CRUD

y almacenar los datos en MySQL. Para el desarrollo del proyecto se utilizó la versión de java 11, packaging jar y el type Maven. Para las dependencias se usaron Spring Boot DevTools, Thymeleaf, Spring Web, Spring Data JPA, H2 Database y MySQLDriver. Las actividades que se desarrollaron fueron varias, entre ellas, la creación de la clase Entidad JPA anotada con @Entity., creación de la clase de acceso a datos DAO con Entity Manager @Repository., la creación de controlador con la acción handler listar., crear la vista html y aplicar estilos CSS Bootstrap. además de implementar Dao con la interface CrudRepository y crear la BD, tabla para los datos en mysql.

Práctica8 Spring MVC & Data JPA: Para esta práctica se continuará el proyecto anterior, con el fin de implementar un paginador para la lista de clientes, agregándole estilos CSS Bootstrap. Las actividades que se desarrollaron fueron varias, entre ellas el uso de Bonus track: layout usando fragments en Thymeleaf, Flash Messenger, paginador desde cero usando PageAndSortingRepository, PageRender y paginador-nav,

Práctica9 Spring MVC & Upload: Para esta práctica se continuará el proyecto anterior, con el fin de implementar el subir imagen en el formulario de registro de clientes. Las actividades que se desarrollaron fueron varias, entre ellas, agregamos campo file en formulario y procesar MultipartFile, agregamos método handler para visualizar la imagen, inicializamos un directorio uploads automáticamente usando CommandLineRunner, creamos la clase UploadFileService y realizamos configuraciones en la BD para tamaños de imagen.

Práctica10 Spring MVC + Data JPA+jQuery: Para esta práctica se continuará el proyecto anterior, con el fin de implementar un sistema de factura al cliente con funciones CRUD. Las actividades que se desarrollaron fueron varias, entre ellas, la creación de asociaciones: ManyToOne Bidireccional, OneToMany Unidireccional de clases Entity factura, cliente e Itemfactura. La creación de controlador de facturaController, crear la plantilla de la lista de facturas por clientes, calcular el total de importe, usar JPA para buscar productos autocomplete. Además de crear crud repository factura para ver, eliminar, guardar, modificar y crear.

Práctica11 Mejorando proyecto: Para esta práctica se continuará el proyecto anterior, con el fin de mejorar el sistema. Las actividades que se desarrollaron fueron la optimización de consulta JPQL en factura con join fetch para obtener los ítems y la optimización de consulta JPQL en clientes con join fetch para obtener las facturas.

Práctica12 Spring Security: Para esta práctica se continuará el proyecto anterior, con el fin de aplicar seguridad en el sistema. La dependencia que se agregara es Thymeleaf Extras Security 4. Las actividades que se desarrollaron fueron varias, entre ellas, la implementación del método BcryptPasswordEncoder, springSecurityConfig para registrar usuarios y agregar roles., método configure httpsecurity para reglas ACL en las rutas. Configuración para el form login y logout. Messages personalizados de acceso al usuario. Crear plantilla login para los usuarios y chequear autorización con securityContextHolderAwareRequest.

Práctica13 Spring Security Autenticación JDBC: Para esta práctica se continuará el proyecto anterior, con el fin de aplicar seguridad en el sistema con la BD. Las actividades que se desarrollaron fueron la modificación del método BcryptPasswordEncoder, la configuración de JDBC authentication y la creación del esquema de tablas BD.

Práctica14 Spring Security Autenticación JPA: Para esta práctica se continuará el proyecto anterior, con el fin de aplicar seguridad en el sistema. Las actividades que se desarrollaron fueron la creación de entidades de usuario y roles, la creación de repositorio JPA. La creación, configuración y registro del servicio JpaUserDetailsService.

Práctica15 Locale y Multilenguaje I18N: Para esta práctica se continuará el proyecto anterior, con el fin de implementar el cambio de idioma en el sistema. Las actividades que se desarrollaron fueron varias, entre ellas, la creación de archivos de idiomas ES, US y DE. La creación del controlador local y links en el menú de navegación para idiomas. Agregar menú desplegable para seleccionar los idiomas y la traducción de textos del paginador.

Práctica16 View Technologies: Para esta práctica se continuará el proyecto anterior, con el fin de implementar el exportar a PDF en el sistema. La dependencia por agregar es OpenPDF. Las actividades que se desarrollaron fueron el diseño de la vista pdf de factura, creación de

link para exportar documento, Traducir la vista PDF a diferentes idiomas y escribir código de facturación PDF.

Práctica17 View Technologies: Para esta práctica se continuará el proyecto anterior, con el fin de implementar el exportar a Excel en el sistema. La dependencia por agregar es Poi-ooxml y poi. Las actividades que se desarrollaron fueron la implementación de la clase vista FacturaXlsxView para la factura, escribir código de detalles de factura en la vista de Excel, Traducir la vista de Excel a diferentes idiomas y mejorar el diseño de platilla Excel.

Práctica18 View Technologies: Para esta práctica se continuará el proyecto anterior, con el fin de implementar el exportar a CSV en el sistema. La dependencia por agregar es super csv. Las actividades que se desarrollaron fueron la configuración y creación de clase vista clienteCsvView y el implemento de la clase clienteCsvView y link en vista en listar clientes.

Práctica19 View Technologies: Para esta práctica se continuará el proyecto anterior, con el fin de convertir documento XML en el sistema. La dependencia por agregar es spring-oxm, Jaxb-API y Jaxb Runtime. Las actividades que se desarrollaron fueron la creación de clase view ClienteListXMLView, implementación de link en vista listar clientes y configuración y creación de clase XMLWrapper.

Práctica20 View Technologies: Para esta práctica se continuará el proyecto anterior, con el fin de convertir a Json en el sistema. Las actividades que se realizaron fueron la creación de la clase view ClienteListJsonView y modificar las clases entities con anotaciones Json para manejar las referencias.

Práctica21 View Technologies: Para esta práctica se continuará el proyecto anterior, con el fin de implementar API REST en el sistema. Las actividades que se realizaron fueron usar el método handler con anotación @ResponseBody y el uso de anotación @RestController sobre la clase controller.

Práctica22 Autenticación basada en JWT: Para esta práctica se duplico el proyecto anterior de web con spring boot, con el fin de crear una autenticación JWT. Las actividades que se desarrollaron fueron varias, entre ellas, proteger las rutas de la Api Rest, crear la clase filtro JwtAuthenticationFilter, generar el JWT, recibir los datos del login en estructura json, validación de token JWT con parse e implementación de la clase MixinGrantedAuthority.

Práctica23 Deployment; Para esta práctica se duplico el proyecto3 anterior de web con spring boot, con el fin de implementar despliegue en servicios externos. Las actividades que se desarrollaron fueron varias, entre ellas, generar archivo jar para despliegue, descargar y modificar nuestro proyecto war que es el servidor externo Tomcat, descargar y generar el war que es el servidor externo WildFly y descargar y generar el war y deploy que es el servidor externo Glassfish.

Práctica24 Proyecto con angular: En esta practica se hizo un proyecto web de spring boot de inventario, con el fin de aplicar todo lo visto en el curso. Para ello se utilizaron los programas Docker que se utilizó como contenedor, plantillas de login y servicios de seguridad, roles, además de alojamiento. Postman para realizar pruebas de conexión con el spring. Spring Boot en el que se trabajó la parte de back-end para productos, categorías y representaciones graficas. visual Studio se usó para la parte de front-end, para las vistas de productos, categorías y graficas. y MySQL Workbench para conexión con spring y guardar la información del proyecto.