

Diseño instruccional de eventos a distancia a través de medios virtuales

Desarrollo Backend con JAVA y el Framework SPRING BOOT

Código: CTI.E.01524/23
Edición 01


JEFATURA
TÉCNICO PEDAGÓGICO
Centro de Capacitación en
Tecnologías de la Información
y las Comunicaciones -TICS-
Intecap
Instituto Técnico
de Capacitación y Productividad
27/02/2023

R.O.DT-08, Edición 02



Guatemala, febrero 2023

Información general del curso

Datos generales

Nivel de competencia	3	Tipo de competencia	Específica	Nivel ocupacional	Medio	Especialidad	Desarrollo de aplicaciones
Área de competencia	Apoyo y soporte administrativo			Subárea de competencia	Desarrollo de sistemas		

Duración 40 h

Objetivo

Durante el curso el participante adquirirá conocimientos para desarrollar aplicaciones web con Java utilizando Spring Boot como Framework, haciendo uso del patrón de dependencias, JPA, Api Rest Full, modelo MVC, aplicando técnicas, procedimientos asertivos y siguiendo las buenas prácticas laborales, de acuerdo con el perfil de egreso del participante.

Destinatarios

Adolescentes y adultos interesados en formarse en temas relacionados bajo la modalidad a distancia por medios virtuales, que cuente con acceso a Internet y un dispositivo electrónico con los requerimientos mínimos para soportar la plataforma, además de tener la disponibilidad de tiempo para ingresar, estudiar las sesiones, realizar las actividades y las evaluaciones previstas para finalizar el curso en el tiempo estipulado.

Requisitos de ingreso del participante

Edad de ingreso: 16 años en adelante
Escolaridad: Tercero básico en adelante
Experiencia: no necesaria
Vinculación laboral: no necesaria
Proceso de admisión: no necesaria

Requisitos del tutor (cuando sea formación grupal)

Experiencia:
Tres años de experiencia como mínimo. Cuando no se cumple con los años de experiencia, indicar o haber aprobado el programa de formación técnica de instructores del INTECAP.

Competencias pedagógicas:

Poseer el Certificado de Formación Metodológica, extendido por el INTECAP o presentar constancia de haber aprobado la formación metodológica, estar cursándola o firmar compromiso de cursarla.

Formación académica:

Título o diploma de educación media. Cuando estos no se relacionen con la especialidad requerida, debe indicarse y acreditar Certificado de Aptitud Profesional o Crédito Ocupacional o Certificado Ocupacional o bien, Título Ocupacional afín a la especialidad.

Vigencia del diseño

El presente plan para la formación tiene una vigencia de 01 años a partir de la fecha de divulgación.

Documento de certificación a expedir

Constancia de participación

Formación vinculada

No aplica

Núm.	Unidades virtuales
1	Desarrolla aplicaciones web con Java utilizando Spring Bot como Framework

Programa de contenidos para la formación por unidades virtuales

Unidad virtual 1: Desarrolla aplicaciones web con Java utilizando Spring Bot como Framework	Duración	40
---------------------------------------------------------------------------------------------	-----------------	-----------

Número y nombre de sesión	Criterios de logro	Dosificación de contenidos	Actividades	Recursos virtuales
Sesión 1: Conceptos generales	<ul style="list-style-type: none"> Identifica los 4 pilares de la programación orientada a objetos. Describe el funcionamiento de una API. Prepara el ambiente de desarrollo e instalando y configurando los programas necesarios 	1. Conceptos Generales 1.1. Roles Backend y Frontend 1.2. Framework 1.3. Pilares de la programación POO 1.3.1. Herencia 1.3.2. Polimorfismo 1.3.3. Encapsulamiento 1.3.4. Abstracción 1.4. Modelo Cliente Servidor 1.4.1. Request 1.4.2. Get 1.4.3. Post 1.4.4. Put 1.4.5. Delete 1.5. Como funciona una API 1.6. ¿Qué es API RESTful? 1.7. Formato Json 1.8. ¿Qué es Spring boot? 1.9. Instalación de Ambiente de Desarrollo 1.9.1. Jdk 11 configuración variable de entorno del sistema 1.9.2. IntelliJ IDEA 1.9.3. MySQL 1.9.4. Postman 1.9.5. Git	Actividad para desarrollar habilidades y destrezas 1: - Instalación y configuración de ambiente de desarrollo - JDK 11 configuración de variables de entorno del sistema - IntelliJ IDEA - Mysql - Postman - Git	Sesión sincrónica presencial a través de Teams. Grabación de la sesión
Sesión 2: Spring MVC	<ul style="list-style-type: none"> Identifica los errores en la configuración del Spring MVC. Describe el Modelo Vista Controlador en un proyecto. Define la estructura de 	2. Spring MVC 2.1. Creando un proyecto web con Spring Boot 2.2. Estructura de una aplicación Spring Boot 2.3. Creando el controlador de y la vista 2.4. Errores de configuración 2.5. Pasar datos a la vista	Actividad para desarrollar habilidades y destrezas 2: - Creación de Proyecto de Spring Boot y despliegue de 10 Registro a través de un modelo de datos usando Thymeleaf y Bootstrap 4	

Número y nombre de sesión	Criterios de logro	Dosificación de contenidos	Actividades	Recursos virtuales
Sesión 3: Inyección de dependencias	<p>una aplicación Spring Boot.</p> <ul style="list-style-type: none"> Describe el funcionamiento de una inyección de dependencias. Diseña un proyecto utilizando servicios RestFull. 	<p>2.6. Antacion @RequestMapping sobre el controlador</p> <p>2.7. Que es el MVC</p> <p>2.8. El objeto Model</p> <p>2.9. Directiva if de Thymeleaf</p> <p>2.10. Directivas each (Foreach) de Thymeleaf</p> <p>2.11. Anotación @ModelAttribute</p> <p>2.12. Anotación @RequestParam</p> <p>2.13. Obtener varios parámetros de la URL con @RequestParam</p> <p>2.14. Anotación @PathVariable</p> <p>2.15. Agregando recursos estilos y estilos CSS personalizado</p> <p>2.16. Retornar Direct y Forward como respuesta en métodos del controlador</p> <p>2.17. Despliegue y ejecución desde terminal (deploy)</p> <p>3. Inyección de Dependencias</p> <p>3.1. Creación de nuevo proyecto y controlador</p> <p>3.1.1. Añadiendo la clase de servicio</p> <p>3.1.2. Implementando inyección de dependencia con la anotación @Autowired</p> <p>3.1.3. Inyectado mediante las interfaces</p> <p>3.1.4. Introducción al contenedor y la inyección de dependencias</p> <p>3.1.5. Inyección de dependencias vía constructor y métodos setters</p> <p>3.1.6. Anotación @Primary</p> <p>3.1.7. Anotación @Qualifier</p> <p>3.1.8. Registrando componentes usando clase de configuración y la anotación @Bean</p> <p>3.1.9. Ejemplo Factura y relaciones entre sus componentes</p> <p>3.1.10.Registrando componente ItemsFactura</p> <p>3.1.11.Creando el controlador FacturaController y su vista</p> <p>3.1.12.Registrando otras lineas para la factura</p> <p>3.1.13.Ciclo de vida del componente</p> <p>3.1.14.. Contextos (Scope) de los componentes</p>	<p>Actividad para desarrollar habilidades y destrezas 3:</p> <ul style="list-style-type: none"> Creación de Servicio RestFull de ítems de factura 	<p>Sesión sincrónica presencial a través de Teams.</p> <p>Grabación de la sesión</p>

Número y nombre de sesión	Criterios de logro	Dosificación de contenidos	Actividades	Recursos virtuales
Sesión 4: Nuevo proyecto con form	<ul style="list-style-type: none"> Describe que es un atributo del objeto model en un formulario. Diseña mensajes de errores personalizados utilizando properties. Identifica las diferentes validaciones en un proyecto de desarrollo utilizando formularios. Personaliza un formulario con estilos CSS. 	<p>4. Nuevo proyecto con form</p> <p>4.1. Actualización: para Spring Boot 2.3.0 o superior agregar dependencia validation</p> <p>4.1.1. Añadiendo la vista del formulario</p> <p>4.1.2. Método handler para recibir y procesar los datos enviados</p> <p>4.1.3. Creando la clase model del formulario</p> <p>4.1.4. Mapeando los campos del formulario al objeto model</p> <p>4.1.5. No olvidar agregar la dependencia spring-boot-starter</p> <p>4.1.6. Validación del formulario usando anotaciones</p> <p>4.1.7. Añadiendo los mensajes de errores en la vista del formulario</p> <p>4.1.8. El atributo object de thymeleaf en el elemento form</p> <p>4.1.9. Mostrar valores de atributos del objeto model en el formulario</p> <p>4.1.10. La anotación @SessionAttributes para mantener los datos durante el ciclo del form</p> <p>4.1.11. La anotación @Size para rangos y @Email para validar correo electrónico</p> <p>4.1.12. Mensajes de errores personalizados usando properties</p> <p>4.1.13. Archivos properties y acentos con UTF-8</p> <p>4.1.14. Validación personalizada usando anotación @Pattern para expresiones regulares</p> <p>4.1.15. Validación personalizada usando anotación @Pattern para expresiones regulares</p> <p>4.1.16. Registrando una clase validador con la anotación @InitBinder</p> <p>4.1.17. Validación personalizada usando anotaciones</p> <p>4.1.18. Validación personalizada usando anotaciones ejemplo requerido</p> <p>4.1.19. Validación de números enteros con @Max @Min y NotNull</p> <p>4.1.20. Validación de fechas con @NotNull y dando formato con @DateTimeFormat</p>	<p>Actividad para desarrollar habilidades y destrezas 4:</p> <ul style="list-style-type: none"> Validación de formulario agregando mensaje personalizados con estilos css y restricción de tipo de datos. 	<p>Sesión sincrónica presencial a través de Teams.</p> <p>Grabación de la sesión</p>

Número y nombre de sesión	Criterios de logro	Dosificación de contenidos	Actividades	Recursos virtuales
Sesión 5: Mostrando data en el vista	<ul style="list-style-type: none"> Envía información por medio del método POST. Implementa el framework de bootstrap en un proyecto. Identifica los componentes de un formulario. Agrega estilos CSS Bootstrap en una plantilla. 	<p>4.1.21. Validación de fechas con input type date de HTML5 y @Past y @Future</p> <p>4.1.22. Formateando fechas con @InitBinder y registrando un CustomDateEditor</p> <p>4.1.23. Formateando fechas con @InitBinder y registrando un CustomDateEditor Parte 2</p> <p>4.1.24. Lista select desplegable</p> <p>4.1.25. Formateando fechas en vistas thymeleaf</p> <p>4.1.26. Implementando propio filtro custom property editor para convertir a mayúscula</p> <p>5. Mostrando data en la vista</p> <p>5.1. Llenando lista select con Map</p> <p>5.2. Llenando lista select con objetos de la clase País</p> <p>5.3. Añadiendo componente service de País</p> <p>5.4. Añadiendo componente pais property editor</p> <p>5.5. Implementando checkboxes</p> <p>5.6. Llenando checkboxes con Map</p> <p>5.7. Llenando checkboxes con objetos del tipo Role</p> <p>5.8. Añadiendo componente role property editor</p> <p>5.9. Checkbox booleano true o false</p> <p>5.10. Radio button</p> <p>5.11. Input type hidden</p> <p>5.12. Poblando pais y roles</p> <p>5.13. Redirect después del POST procesar</p> <p>5.14. Añadiendo estilos CSS al formulario con Bootstrap</p> <p>5.15. Añadiendo estilos CSS Bootstrap a la plantilla resultado</p> <p>5.16. Cargando archivo css Bootstrap localmente</p>	<p>Actividad para desarrollar habilidades y destrezas 5:</p> <ul style="list-style-type: none"> Enviar información a través del método POST con planilla css y Bootstrap comprobar servicio utilizando postman. 	<p>Sesión sincrónica presencial a través de Teams.</p> <p>Grabación de la sesión</p>
Sesión 6: Interceptores HTTP	<ul style="list-style-type: none"> Describe los interceptores HTTP. Implementa interceptores de tiempo transcurrido. 	<p>6. Interceptores HTTP</p> <p>6.1. Introducción a los interceptores Http</p> <p>6.2. Implementando interceptor Tiempo Transcurrido</p>	<p>Actividad para desarrollar habilidades y destrezas 6:</p> <ul style="list-style-type: none"> Creación de proyecto utilizando interceptores con peticiones post 	

Número y nombre de sesión	Criterios de logro	Dosificación de contenidos	Actividades	Recursos virtuales
Sesión 7: Creando un nuevo proyecto	<ul style="list-style-type: none"> Utiliza Postman para verificar la funcionalidad de los servicios. Identifica el método handler cerrado. Describe el error NullPointerException. 	6.3. Registrando interceptor en la configuración de spring 6.4. Implementando fragment interceptor para reutilizar código en thymeleaf 6.5. Saliendo del interceptor en peticiones POST http - return en post handle 6.6. Retornando false en un interceptor 7. Creando nuevo proyecto 7.1. Creando interceptor horario 7.2. Creando método handler cerrado 7.3. Registrando interceptor 7.4. Revisando error NullPointerException	<ul style="list-style-type: none"> Utilizar postman para verificar el response de servicio. <p>Actividad para desarrollar habilidades y destrezas 7:</p> <ul style="list-style-type: none"> Creación de Horario de acceso a cliente para registrarse a una suscripción en un rango de Hora. 	Sesión sincrónica presencial a través de Teams. Grabación de la sesión
Sesión 8: Manejo de Errores	<ul style="list-style-type: none"> Identifica los errores mas frecuentes en el ambiente de desarrollo. Describe la clase Exception. Utiliza Try Catch para el manejo de errores. 	8. Manejo de Errores 8.1. Creando nueva aplicación y controlador 8.2. Personalizando error 404 y 500 8.3. Anotación @ControllerAdvice para manejo de excepciones 8.4. Personalizando página de error para la excepción NumberFormatException 8.5. Manejando excepción personalizada creando clases Model y Service 8.6. Añadiendo método handler 8.7. Creando clase Exception personalizada 8.8. Lanzamiento de excepción usando Api Optional de Java 8	<p>Actividad para desarrollar habilidades y destrezas 8:</p> <ul style="list-style-type: none"> Creación de template para páginas de errores 404 y 500 debe agregar las excepciones según sea el caso cada lanzamiento se debe integrar el Try.. Catch 	
Sesión 9: Spring MVC & Data JPA	<ul style="list-style-type: none"> Utiliza estilos Bootstrap para diseñar un formulario. Construye vistas en un proyecto. Describe la consola H2 Database. Implementa diferentes mensajes de error para el usuario en un formulario. 	9. Spring MVC & Data JPA: Desarrollo de API 9.1. Creando el proyecto 9.2. Creando la clase Entidad JPA anotada con @Entity 9.3. Creando la clase de Acceso a Datos o DAO con Entity Manager - @Repository 9.4. Creando controlador con la acción handler listar 9.5. Creando la vista listar.html	<p>Actividad para desarrollar habilidades y destrezas 9:</p> <ul style="list-style-type: none"> Creación de la base de datos de AgendaEstudiantes, crear la clase DAO, crear repositor y configurar la información en propiedades para hacer la consulta de información en Mysql. 	

Número y nombre de sesión	Criterios de logro	Dosificación de contenidos	Actividades	Recursos virtuales
Sesión 10: Spring MVC & JPA: Buenas prácticas de aplicación	<ul style="list-style-type: none"> ▪ Crea un servicio para paginación de datos. ▪ Utiliza el Framework de Bootstrap para tablas de registros. ▪ Identifica las buenas prácticas de aplicación de Spring Boot. 	<p>9.6. Decorando la vista listar con estilos CSS Bootstrap</p> <p>9.7. La consola H2 Database</p> <p>9.8. Agregando crear en Repositorio Dao y Controlador</p> <p>9.9. Creando la vista crear y formulario</p> <p>9.10. Mejorando formulario con estilos Bootstrap CSS</p> <p>9.11. Agregando campo fecha en formulario</p> <p>9.12. Actualización: para Spring Boot 2.3.0 o superior agregar dependencia validation</p> <p>9.13. Agregando reglas de validación usando anotaciones</p> <p>9.14. Agregando mensajes de errores personalizados</p> <p>9.15. Agregando mensajes de errores sobre el formulario</p> <p>9.16. Agregando editar</p> <p>9.17. Agregando eliminar</p> <p>9.18. Agregando la clase Service</p> <p>9.19. Implementando el Dao con la interface CrudRepository</p> <p>9.20. Cambiando Datasource a MySQL en el proyecto</p> <p>10. Spring MVC & JPA: Buenas prácticas de aplicación</p> <p>10.1. Layout usando fragments en Thymeleaf</p> <p>10.2. Layout usando fragments en Thymeleaf - Parte II</p> <p>10.3. Flash Messenger</p> <p>10.4. Paginador desde cero - Usando PagingAndSortingRepository - Parte I</p> <p>10.5. Paginador desde cero - La clase PageRender y controlador - Parte II</p> <p>10.6. Paginador desde cero - La vista fragmento paginator-nav - Parte III</p> <p>10.7. Paginador desde cero - Agregando estilos Bootstrap - Parte IV</p>	<p>Actividad para desarrollar habilidades y destrezas 10: Evaluación</p> <ul style="list-style-type: none"> - Crear servicio para la paginación de datos utilizando el Framework de Bootstrap y la tabla de registro de pacientes. - Crear todas las router, controller y el mapeo de datos. 	<p>Sesión sincrónica presencial a través de Teams.</p> <p>Grabación de la sesión</p>

Lista de dotación (cuando aplique)

Materiales	Maquinaria y equipo	Herramientas/utensilios/equipo de protección personal
<ul style="list-style-type: none"> ▪ Contenidos virtuales ▪ Documentos de apoyo 	<ul style="list-style-type: none"> ▪ Equipo de cómputo con características mínimas: <ul style="list-style-type: none"> - Disco duro de 500 Gigabytes o superior - Memoria RAM de 4 Gigabytes o superior - Procesador Intel™ i3 o superior - Auriculares 	<ul style="list-style-type: none"> ▪ Internet ▪ Correo electrónico ▪ Audífonos o bocinas ▪ Software <ul style="list-style-type: none"> - Sistema operativo Windows, GNU/Linux o Mac OS - Microsoft Teams

Referencias bibliográficas

Páginas web consultadas:

1. © 2020 Spring by VMWare Tanzu [Todos los derechos reservados]. Disponible en: <https://spring.io/>