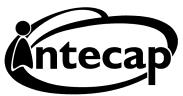
**INSTITUTO TÉCNICO DE CAPACITACIÓN Y PRODUCTIVIDAD DIVISIÓN TÉCNICA**

R.O.DT-08, Edición 01

Nombre: Excel Básico Código: DI.3.11.3-.00000/20 Página 1 de 6



Diseño instruccional de eventos a distancia a través de medios virtuales

**Diseño de interfaces web con REACT JS**

# Código: CTI.E.01525/23

Edición 01



R.O.DT-08, Edición 01

Guatemala, febrero de 2023

Información general del curso

Datos generales

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nivel de competencia | 3 | Tipo de competencia | Especifica | 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 capacidades para desarrollar aplicaciones web SPA utilizando la libreria de React JS haciendo uso de Hooks, Redux, Context API y Firebase, 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 diseño instruccional tiene una vigencia de 1 año a partir de la fecha de divulgación. |

|  |
| --- |
| Documento de certificación a expedir |
| Constancia de aprobación. |

|  |
| --- |
| Formación vinculada |
| No aplica. |

|  |  |
| --- | --- |
| Núm. | Unidades virtuales |
| 1 | Diseña interfaces web utilizando la librería React JS |

Programa de contenidos para la formación por unidades virtuales

|  |  |  |
| --- | --- | --- |
| Unidad virtual 1: Diseña interfaces web utilizando la librería React JS | Duración | 40 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Número y nombre de sesión | Criterios de logro | Dosificación de contenidos | Actividades | Recursos virtuales |
| Sesión 1: Conceptos y vocabulario  Sesión 2: React  Sesión 3: Estilos CSS en React  Sesión 4: Administrador de Pacientes  Sesión 5: Control de presupuesto en React  Sesión 6: Aplicación de lista de tarea | * 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 * Identifica los errores en la configuración del Spring MVC. * Describe el Modelo Vista Controlador en un proyecto. * Define la estructura de una aplicación Spring Boot. * Describe el funcionamiento de una inyección de dependencias. * Diseña un proyecto utilizando servicios RestFull. * 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. * Envía información por medio del método POST. * Implementa el framework de boostrap en un proyecto. * Identifica los componentes de un formulario. * Agrega estilos CSS Boostrap en una plantilla. * Describe los interceptores HTTP. * Implementa interceptores de tiempo transcurrido. * Utiliza Postman para verificar la funcionalidad de los servicios. | 1. Conceptos y vocabulario  1.1. Rol de FrontEnd  1.1.1. Roles Backend y Frontend  1.1.2. Framework  1.1.3. Modelo Cliente Servidor  1.1.4. Request  1.1.5. Get  1.1.6. Post  1.1.7. Put  1.1.8. Delete  1.1.9. Como funciona una API  1.1.10. ¿Qué es API RESTful?  1.1.11. Formato Json  1.2. Instalación de Ambiente de trabajo  1.2.1.1. Nvm manejador de paquetes  1.2.1.2. Instalación de node.js  1.2.1.3. Instalación de Git  1.2.1.4. Instalación de Postman  1.2.1.5. Instalación de Vscode  1.2.1.6. Instalación de Extensiones  1.3. Repaso de fundamentos de JavaScript  1.3.1. Variables, estructura y tipo de datos  1.3.2. Objetos, destruturing de 2 o mas objetos  1.3.3. Unir objetos  1.3.4. Array, manipulación, destructuring, e iteraciones  1.3.5. Funciones, funciones con parámetros, función flecha  1.3.6. Condiciones if, comparaciones , or y and  1.3.7. Condiciones ternarios  1.3.8. Scope de variable  1.3.9. Slectores del DOM – Query Selector  1.3.10. Manipular el Html con Js  1.3.11. Eventos Dom Clicks, inputs, submit  1.3.12. Generación de Código Html con JS  1.3.13. Import y exports  1.3.14. Fetch api y promises  1.3.15. Fetch Api con Async Await  1.3.16. Múltiples Async / Await Performace  1.3.17. Template String  2. React  2.1. ¿Qué es react?  2.2. Introducción a los componentes en React  2.3. Componentes en múltiples archivos  2.4. Propiedades de los componentes  2.5. Estado de los Componentes  2.6. Como trabajar con formularios  2.7. Formulario y estados  2.8. Componentes basados en clases  2.9. Como convertir componentes basados en clase a componentes funcionales  2.10. Métodos de ciclo de vida (Life Cycle Methods)  2.11. Como integrar la terminal en Visual Studio Code (Windows)  3. Estilos CSS en React  3.1. Como trabajar con estilos CSS dentro de React  3.2. Módulos de CSS  3.3. Styled Components  4. React Hooks  4.1. Hook de useState  4.2. Hook de useEffect  4.3. Ejemplo useEffect  4.4. Hook de useReducer  4.5. Hooks personalizdos  4. React Hooks  4.1. Hook de useState  4.2. Hook de useEffect  4.3. Ejemplo useEffect  4.4. Hook de useReducer  4.5. Hooks personalizdos  5. Administrador de Pacientes  5.1. Análisis del proyecto  5.2. Estructura básica del proyecto  5.3. Qué es jsx  5.4. JavaScript en Html con Jsx  5.5. Crando y llamando a nuestro componente  5.6. React developer Tools  5.7. Forma de Escribir código Css en React  5.8. Instalando Tailwind Css  5.9. Añadiendo la primeras clase Tailwind  5.10. Aplicando Clases al header  5.11. Creando el componente de formulario  5.12. Creado un input en el formulario  5.13. Campos de formulario  5.14. Listado de Pacientes agregando Hooks  5.15. State en React  5.16. Reglas de Hooks  5.17. Leyendo la información escrita en un Input y colocandola en el State  5.18. Eventos y Sintaxis en React  5.19. Leyendo el Resto de los Inputs y colocandolos en el State  5.20. Validación de Formularios en React  5.21. . Mostrar un mensaje de error  5.22. Pasando información entre Componentes con Props  5.23. Añadiendo Pacientes al State principal y reiniciando Formularios  5.24. Creando un Componente de Errores  5.25. El prop {children}  5.26. iterando sobre un Array en React  5.27. Mostrando la Información de los pacientes en el listado  5.28. Generar un ID para cada paciente  5.29. Mostrar Textos de Forma condicional en React  5.30. Creando botones de Editar y eliminar  5.31. Registrando un evento en el Botón de Editar  5.32. El Hook useEffect y Reto 1  5.33. SOLUCIÓN a Reto 1  5.34. Colocando en el State el paciente a editar  5.35. Mostrando e Forma condicional Textos en el Formulario  5.36. Detectar un Registro Nuevo o una edición  5.37. Eliminar un Paciente  5.38. Agregando LocalStorage para los Pacientes  5.39. . Finalizando la funcionalidad de LocalStorage  5.40. Deployment de un Proyecto en React  5.41. Deployment por medio de Git  5.42. Añadiendo Integración Continua  6. Control de Presupuesto en React con Vite y Local Storage 18/03/2023  6.1. Iniciando el proyecto e instalación de dependencias  6.2. Creando Componente para presupuesto  6.3. Creando el Formulario de Presupuesto  6.4. Validando el Presupuesto  6.5. Mostrando el Componente de Gastos si el presupuesto es válido  6.6. Primeros Pasos con el Componente de Control de Presupuesto  6.7. Formatear Números a Dinero  6.8. Añadir un botón para registrar un nuevo gasto  6.9. Primeros pasos para crear un Modal  6.10. Mostrando y Ocultando el Modal  6.11. Creando una Animación en el Modal  6.12. Añadiendo el Formulario de Gastos  6.13. Creando el State para los gastos  6.14. Validando los Gastos  6.15. Creando una función para generar un ID único y Añadiendo Gastos al State  6.16. Creando el Componente para Crear un Gasto  6.17. Mostrando los Gastos  6.18. Formateando las Fechas  6.19. Mostrando el Icono de Gastos  6.20. Calculando el Presupuesto Gastado  6.21. Calculando el Presupuesto Disponible  6.22. Añadiendo un Efecto de Swipe  6.23. Finalizando el Efecto de Swipe  6.24. Continuación del proyecto Presupuesto  6.25. Detectando el Gasto a Editar al dar Swipe  6.26. Colocando el Gasto a Editar en el State y en el Formulario  6.27. . Editando los Gastos  6.28. Eliminar Gastos con Swipe  6.29. Resetear el State al finalizar un gasto editado  6.30. Agregando una Gráfica Circular para el control del presupuesto  6.31. Calcular el Porcentaje gastado para la gráfica Circular  6.32. Personalizar la gráfica Circular con Estilos  6.33. Mostrando Textos en Gráfica Circular  6.34. Colocar el Presupuesto en LocalStorage  6.35. Colocar los Gastos en Local Storage  6.36. Creando un Filtro de Categorías de Gastos  6.37. Colocando Gastos Filtrados en el State  6.38. Mostrando Gastos Filtrados  6.39. Cambiando la apariencia de Presupuesto Disponible si ya fue agotado  6.40. Añadiendo un botón para reiniciar la app  7. Creando una aplicación de Lista de Tarea  7.1. Análisis del Proyecto  7.2. Creando el encabezado de la App  7.3. Formulario para agregar Tareas  7.4. Creando la lista de tareas  7.5. Formulario Editar Tarea  7.6. Función para completar tarea  7.7. Editar y Borrar tarea  7.8. Lógica para mostrar y ocultar las tareas completadas  7.9. Agregando localstorage a nuetra app  7.10. Publicar tarea a Railway  7.11. Subir repositorio a Git | Instalación de Ambiente de Trabajo   1. Vscode y Plugins 2. Git 3. Postman 4. Nvm manejadores de paquetes 5. Carrito de compra en consola utilizando el dom.   SPA en react de perfil de usuario  Y estilos css  Formulario de Login Básico  Consumo de The Rick and Morty API  Proyecto de Registro de Paciente IGSS utilizando tailwindcss con localstore  Creación de Proyecto utilizando Vite, creación de efectos para gráficos para el presupuesto y vistas de componente  Creación de nuevo proyecto y realización de vista de proyecto utilizando localStorage, validación de formulario de creación de tareas. | Sesión sincrónica  presencial a través de Teams.  Grabación de la sesión  Sesión sincrónica  presencial a través de Teams.  Grabación de la sesión    Sesión sincrónica  presencial a través de Teams.  Grabación de la sesión  Sesión sincrónica  presencial a través de Teams.  Grabación de la sesión  Sesión sincrónica  presencial a través de Teams.  Grabación de la sesión |

Lista de dotación (cuando aplique)

|  |  |  |
| --- | --- | --- |
| Materiales | Maquinaria y equipo | Herramientas/utensilios/ equipo de protección personal |
| * Contenidos virtuales * Documentos de apoyo | * Equipo de cómputo con características mínimas: * Disco duro de 500 Gigabytes o superior * Memoria RAM de 4 Gigabytes o superior * Procesador IntelTM i3 o superior * Auriculares | * Internet * Correo electrónico * Audífonos o bocinas * Software * Sistema operativo Windows, GNU/Linux o Mac OS * Acceso a plataforma AWS Academy |

Bibliografía base del diseño instruccional

Páginas web consultadas:

1. © 2023 React Documentation [Todos los derechos reservados]. Disponible en: https://es.reactjs.org/docs/getting-started.html