2022

Imagina

CURSO EN DIRECTO A MEDIDA —

Buenas Prácticas en Desarrollo Web con React

Descripción del curso

Introducción

Aprende las mejores prácticas para desarrollar software de forma correcta y eficaz. Explora conceptos como patrones de diseño, principios SOLID, metodologías ágiles o código limpio.

Objetivos

- Comprender las fases por las que pasa un desarrollo software
- Gestionar las versiones de SW con GIT y aprender buenas prácticas en la gestión de repositorios y ramas con Gitflow
- Ser capaces de identificar que metodología se adapta más a nuestras necesidades
- Realizar tests unitariosn con Jest y funcionales con Cypress
- Conocer y asimiliar conceptos que nos ayuden a generar un código de mayor calidad y mejor documentado

¿A quién va dirigido?

Todos aquellos desarrolladores que quieran asentar unas buenas bases en cuanto a buenas prácticas para sus desarrollos

Requisitos

- Tener acceso a un equipo con permisos de instalación
- Tener instalado Node LTS, NPM y Git en su última versión en el equipo
- Tener instalado previamente Visua Studio Code y Fork (https://git-fork.com/), en sus últimas versiones, en el equipo
- Tener conocimientos desarrollando con JS, HTML, CSS y SCSS y haber desarrollado aplicaciones React
- Se recomienda tener una cuenta de Github antes de la formación

Horario del curso

Aquí te mostramos el horario en **hora peninsular** de las sesiones formativas con los respectivos temarios por cada sesión, en las siguientes páginas podrás revisar el contenido de los temas que se indican.

SESIÓN 1

Día 13 de junio

12:00 a 14:00

- Tema 1
- Tema 2
- Tema 3

SESIÓN 2

Día 16 de junio

12:00 a 14:00

- Tema 4
- Tema 5

SESIÓN 3

Día 20 de junio

12:00 a 14:00

- Tema 6
- Tema 7
- Tema 8

SESIÓN 4

Día 22 de junio

12:00 a 14:00

- Tema 9
- Tema 10
- Tema11

SESIÓN 5

Día 27 de junio

12:00 a 14:00

- Tema 12
- Tema 13

Temario del curso

1 METODOLOGÍAS ÁGILES

- Las metodologías ágiles
- El manifiesto ágil
- Scrum
- Kanban
- Peer review

2 TRABAJO EFICIENTE CON GIT

- Configuración de GIT
- Herramientas visuales para trabajar en Visual Studio Code
- Fork como aplicación recomendada y otras alternativas
- Ver el historial de confirmaciones
- Etiquetas (tagging)
- Deshacer cambios
- Gestión de Ramas
- Rebase
- Stash
- Cherry Pick
- Forking workflow
- Protocolo 2Phase commit
- GitHooks
- Archivo GitIgnore

3 GESTIÓN DE CONFLICTOS EN GIT

- Cómo se generan los conflictos
- Resolución de conflictos desde VSCode y desde Fork
- Recuperar versiones anteriores
- Revisiones e historia de documentos

4 GITFLOW

- ¿Qué es el desarrollo paralelo?
- ¿Qué es Gitflow
- Instalación
- Estrategias de branching automáticas con GitFlow
- Comandos esenciales

5 TESTING Y TDD

- Introducción al testing unitario y de integración
- El concepto del Coverage y la importancia de éste para garantizar la robustez calidad
- Introducción a la metodología TDD

Temario del curso

- Usando Jest para hacer pruebas unitarias a componentes
- ¿Qué se debe probar?
- Creación de tests unitarios
- Empleando Mocks, Spies y Stubs de Jest para simular ejecución de funciones y peticiones http
- Instalación e introducción a Cypress
- API de Cypress y creación de pruebas
- Automatización CI/CD con Github actions

6 PRINCIPIOS SOLID

- ¿Qué son?
- Principio de la responsabilidad única
- Principio de abierto/cerrado
- Principio de la substitución de Liskov
- Principio de segregación de interfaces
- Principio de inversión de la dependencia

7 REFACTORING

- ¿En qué consiste el refactoring?
- Beneficiones de la refactorización de código
- Técnicas de refactorización

8 PATRONES DE ASINCRONÍA Y GESTIÓN DE PETICIONES

- Introducción
- Patrones más destacables

9 INTRODUCCIÓN AL CÓDIGO LIMPIO

- Introducción al concepto de código limpio y su importancia
- ¿Qué se entiende como código incorrecto?
- Las consecuencias de un código incorrecto en términos económicos
- La importancia de la actitud frente al cambio
- Los conceptos del código limpio
- Las diferencias teóricas del concepto
- La Regla de Boy Scout y los principios del código limpio
- Garantizando calidad de código: ESLint en proceso CI

10 LA IMPORTANCIA DEL NOMBRADO

- Técnicas de nombrado limpio para el trabajo en equipo efectivo
- Ejemplos de nombrado incorrecto y cómo resolverlos

Temario del curso

11 LAS FUNCIONES EN EL CÓDIGO LIMPIO

- Técnicas para estructura y garantizar funcionalidad única de las funciones
- Una función, una funcionalidad
- Cómo plantear de forma limpia los argumentos en las funciones
- Los argumentos de indicador
- Verbos y las palabras clave
- Los argumentos de salida de una función
- gestión limpia de excepciones y los bloques Try/Catch
- DRY, No te repitas
- Creación de funciones limpias
- La programación estructurada y la programación funcional
- Ejemplos de funciones incorrectas y cómo resolverlos bajo estándares limpios

12 DOCUMENTANDO A TRAVÉS DE COMENTARIOS

- La importancia de los comentarios
- Los comentarios no pueden excusar un código incorrecto
- ¿Qué es un comentario de calidad?
- Comentarios TODOs y FIXMEs
- Comentarios Informativos y legales
- ¿Qué es un comentario de mala calidad?
- Comentarios confusos, redundantes, mal descritos y sobrantes
- Comentarios periódicos
- Comentarios obligatorios
- Marcadores de posición
- Uso de funciones o variables sin comentarios
- Comentarios como encabezados de funciones
- DocumentJS y otras alternativas de documentación
- Usando Flow (https://flow.org/) para tipar código JS

13 FORMATO DEL CÓDIGO

- ¿Qué es el formato y qué funcionalidad tiene?
- Introducción al formato vertical
- La densidad y distancia vertical en el código
- Formato en la gestión de variables

Temario del curso

- Verticalidad entre funciones dependientes y términos de afinidad
- La importancia del orden vertical
- Introducción al formato horizontal
- La densidad horizontal en el código
- La alineación horizontal
- La indentación o sangrado horizontal
- ¿Qué son los ámbitos ficticios?
- Reglas de equipo y formato imprescindibles con Prettier