



Full-Stack Web Development Bootcamp

ÍNDICE GENERAL

Módulo 1: Introducción al Desarrollo Web

Introducción

HTML

CSS

HTML5

CSS3

Utilidades y otras herramientas – SASS, SCSS, LESS

Bootstrap

Módulo 2. Git & GitHub: Intro

Sistema de control de versiones (SCV o VCS)

Git

Herramientas

GitHub

Módulo 3. Introducción a la programación

Sobre la Programación

Organización de un programa informático

Javascript como primer lenguaje de programación

Variables y tipo de datos

Estructuras de control

Operadores

Excepciones

Funciones y Procedimientos

Módulo 4. Programación & Algorítmica & Proyectos Software

Lenguajes de Programación

Organización de un programa informático avanzado

Entornos

Programación Orientada a Objetos

Algorítmica

Debugging

Programación Funcional: Visión

Planificación de proyectos software



Módulo 5. Introducción a BBDD

Tipos de BBDD
Sistemas Gestores de BBDD
Introducción al uso de IDE para trabajar con MySQL
Fundamentos de SQL
Técnicas para la modificación de datos
Técnicas para la obtención de datos
Generación y uso de secuencias

Módulo 6. Bases de Datos avanzadas

Consultas avanzadas
Trabajando con cadenas
Gestión de duplicados
Estructuras de control
Transacciones
Procedimientos almacenados
Funciones
Índices
Estrategias de optimización

Módulo 7. Git & GitHub: Avanzado

Git
GitHub
GitFlow

Módulo 8. Javascript avanzado & EcmaScript 6

Curiosidades del texto
Propiedades de datos y propiedades de acceso
Alcance - Scope y Closure
Por valor versus por referencia
Colecciones y conjuntos
ES6

Módulo 9. JQuery

Introducción a jQuery
Selectores y el DOM
jQuery y llamadas AJAX



Módulo 10. Node.JS & Express.JS

Introducción

Core de NodeJS

Core de NodeJS (II)

Desarrollo de API usando Express

Desarrollo de API/Microservicio usando boilerplate ES6-Express (Microservicio1)

Desarrollo de API/Microservicio usando boilerplate TypeScript (Microservicio2)

Desarrollo de API/Microservicio usando NestJS (Microservicio3) conectado a PostgreSQL

Testing en Node

Módulo 11. Introducción a Docker

Qué es Docker

Para qué sirve Docker

Instalación

Docker Images

Docker Containers

Docker Volumes

Módulo 12. MongoDB

Introducción a MongoDB

Modelado de datos con MongoDB

Sintaxis de consulta MongoDB

Mongoose

CRUD con MongoDB, NodeJS y Express

Módulo 13. Diseño de APIs.REST

Introducción

¿Qué es REST?

Arquitectura REST

URIs

HTTP

Autenticación

Ejercicio Final

Módulo 14. React & Hooks

React

Introducción a React

Primer proyecto en React

Interactividad en React

Fetch y llamadas HTTP con React

Comunicación de alto nivel

Tipos de componentes



Extras

ReactRouter

Introducción a ReactRouter

ReactRouter Avanzado

Arquitectura de una tienda online

Inicio del proyecto

Comunicación con el servidor

Hooks

Módulo 15. Full Stack: de 0 a producción

Introducción

Diseño de producto

Prototipado

Empezar un proyecto

Desarrollo

Entregas

Producción

Proyecto-Ejercicio Final

Módulo 16. Gestión de proyectos: Scrum

Introducción

Lo tradicional

Gestión ágil

Kanban

XP Programming

Introducción a Scrum

Scrum: Roles y Artefactos

Scrum: Ciclo

Scrum: Buenas prácticas

Proyecto

Módulo 17. Calidad de Software: QA

Introducción

Conceptos de Calidad del Software

Habilidades de un QA

Diseño y ejecución de pruebas manuales

Reporte de fallos

Pruebas automáticas de UI con Cypress

Pruebas de API con Postman

Pruebas Unitarias con Jest



Módulo 1. Introducción al desarrollo web

1. Introducción

¿Qué vas a conseguir en este módulo?

1. Desarrollar un portal web, una vez entendidos cada uno de los lenguajes de programación que se utilizan a nivel de servidores y clientes (navegador web).
2. Aprender a escribir código HTML para estructurar nuestras páginas web.
3. Dar estilos a nuestras páginas con lenguaje CSS (hojas de estilos).
4. Interactuar con elementos HTML a través de Javascript con el DOM.

2. HTML

- 2.1. Introducción
- 2.2. Etiquetas HTML
- 2.3. Entornos de desarrollo
- 2.4. Estructura de un documento HTML
- 2.5. Etiquetas Charset y Title
- 2.6. Párrafos y saltos de líneas y Encabezados
- 2.7. La etiqueta span
- 2.8. Comentarios
- 2.9. Etiquetas de texto asociadas al tipo de letra
- 2.10. Etiquetas de texto asociadas al tipo de información
- 2.11. Enlaces
- 2.12. Declaración del <!DOCTYPE> HTML
- 2.13. Recursos
- 2.14. Tipos de referencias
- 2.15. Ejercicio de rutas
- 2.16. Otros tipos de enlaces
- 2.17. Imágenes
- 2.18. Enlaces e imágenes
- 2.19. Tablas
- 2.20. Listas
- 2.21. Formularios
- 2.22. La etiqueta div



3. CSS

- 3.1. Qué es CSS
- 3.2. Estilos por defecto de los navegadores
- 3.3. Sintaxis del CSS
- 3.4. Tipos de CSS
- 3.5. Tipos de selectores
- 3.6. Ejemplo de selectores en código
- 3.7. Cascada
- 3.8. Herencia
- 3.9. Tipografías del sistema
- 3.10. Formateo de textos
- 3.11. Modelo de caja
- 3.12. Tamaños. Unidades de medida
- 3.13. Sintaxis de margin y padding (top, right, bottom y left)
- 3.14. Colores
- 3.15. Borders
- 3.16. Backgrounds
- 3.17. Position, z-index, float y clear y display
- 3.18. Overflow auto
- 3.19. Ejemplos de layouts
- 3.20. Referencia a propiedades CSS

4. HTML5

- 4.1. Introducción
- 4.2. Navegadores que soportan HTML5
- 4.3. La etiqueta <!DOCTYPE html>
- 4.4. Charset
- 4.5. El significado semántico de etiquetas
- 4.6. Nuevas etiquetas de valor semántico y estructural
- 4.7. Secciones y estructuras
- 4.8. Audios y videos
- 4.9. Formularios
- 4.10. Nuevas APIs HTML5

5. CSS3

- 5.1. Introducción
- 5.2. Diseño web responsive
- 5.3. El ViewPort
- 5.4. Gridview
- 5.5. Media Querys (Maquetación responsive)
- 5.6. Imágenes responsive
- 5.7. Textos responsive
- 5.8. Herramientas de trabajo responsive
- 5.9. Cómo estructurar el trabajo
- 5.10. Transparencias, degradados y sombras
- 5.11. Transformaciones y transiciones
- 5.12. ResetCSS

6. Utilidades y otras herramientas – SASS, SCSS, LESS

- 6.1. Hojas de estilos dinámicas: Sass y Scss
- 6.2. SASS, SCSS en profundidad: Declaraciones y Directivas
- 6.3. Utilizando tipografías de Google Fonts
- 6.4. Utilizando iconos Font Awesome
- 6.5. Bootstrap Framework: Descarga, integración, customización e introducción a sus principales componentes

7. Bootstrap

- 7.1. Introducción y descarga
- 7.2. Navegadores y dispositivos compatibles
- 7.3. Complementos de Javascript
- 7.4. Theming de bootstrap con variables Sass ya integradas
- 7.5. Accesibilidad
- 7.6. Diseño
 - 7.6.1. Containers
 - 7.6.2. Responsive breakpoints
 - 7.6.3. Z-Index
 - 7.6.4. Grid
 - 7.6.5. Utilidades para el diseño
- 7.7. Componentes
 - 7.7.1. Alertas
 - 7.7.2. Badge
 - 7.7.3. Breadcrumb
 - 7.7.4. Botones
 - 7.7.5. Cards
 - 7.7.6. Carousel
 - 7.7.7. Collapse
 - 7.7.8. Dropdowns



- 7.7.9. Formularios
- 7.7.10. Grupo de Inputs
- 7.7.11. Jumbotron
- 7.7.12. Grupo de Listas
- 7.7.13. Media object
- 7.7.14. Modal
- 7.7.15. Navs
- 7.7.16. Navbar
- 7.7.17. Paginación
- 7.7.18. Popovers
- 7.7.19. Progreso
- 7.7.20. Scrollspy
- 7.7.21. Spinners
- 7.7.22. Toasts
- 7.7.23. Tooltips
- 7.8. Utilities



Módulo 2. Git & GitHub: Intro

¿Qué vas a conseguir en este módulo?

Al finalizar los dos módulos de Git & GitHub, seremos capaces:

- Crear un repositorio en GitHub (u otro repositorio Git) y subir tu trabajo
- Manejar los cambios en un proyecto de una forma ordenada y segura
- Colaborar con otros integrantes de un equipo para trabajar juntos en un proyecto
- Resolver conflictos entre versiones
- Usar de forma fluida GitKraken y de forma básica la consola de Git

Se irán intercalando prácticas y ejercicios en todos los puntos para afianzar los conceptos. Al

1. Sistema de control de versiones (SCV o VCS)

- 1.1 ¿Porqué?
- 1.2 ¿Qué es?
- 1.3 Tipos: exclusivos / colaborativos, centralizados / distribuidos
- 1.4 Conceptos básicos
- 1.5 Uso de la consola

2. Git

- 2.1 Repositorio
- 2.2 Commit
- 2.3 Log
- 2.4 Etiqueta o tag. Release
- 2.5 Conflicto
- 2.6 Clonado
- 2.7 Stagingarea

3. Herramientas

- 3.1 Consola o CLI
- 3.2 UI: GitKraken, SourceTree, TortoiseGit...

4. GitHub

- 4.1 Estructura básica de un repo
- 4.2 README.md y markdown



Módulo 3. Introducción a la programación

¿Qué vas a conseguir en este módulo?

En la introducción a la programación aprenderás a entender y aplicar los conceptos fundamentales y las partes imprescindibles de la programación que necesitas para continuar el itinerario del Bootcamp, y convertirte en un programador. Todo en un modo muy práctico con ejemplos y ejercicios.

1. Sobre la Programación

- 1.1. ¿Qué es un programa?
- 1.2. ¿Qué es un lenguaje de programación?
- 1.3. ¿Por qué crear programas informáticos?
- 1.4. ¿Cómo convertirse en un programador?

2. Organización de un programa informático

- 1.1. Estructura básica
- 1.2. Puntos de entradas
- 1.3. Procedimientos y Funciones

3. Javascript como primer lenguaje de programación

- 2.1. Porqué empezamos usando Javascript
- 2.2. Ventajas
- 2.3. Javascript como único lenguaje: Full Stack
- 2.4. Javascript ES5
- 2.5. Javascript ES6

4. Variables y tipo de datos

- 3.1. Variables
- 3.2. Numéricos
- 3.3. Texto
- 3.4. Literales
- 3.5. Objetos
- 3.6. Arrays, Listas y Diccionarios

5. Estructuras de control

- 4.1. Sentencia condicional
- 4.2. Iteración y Bucles
- 4.3. Indirección



6. Operadores

- 5.1. Introducción a operadores
- 5.2. Operadores Aritméticos
- 5.3. Operadores de Strings
- 5.4. Otros Operadores

7. Excepciones

- 6.1. Qué son las excepciones
- 6.2. Manejando excepciones
- 6.3. Lanzando excepciones

8. Funciones y Procedimientos

- 7.1. Funciones
- 7.2. Procedimientos
- 7.3. Funciones en Javascript



Módulo 4. Programación & Algorítmica & Proyectos Software

1. Lenguajes de Programación

- 1.1. Paradigmas de programación
- 1.2. Programación orientada a objetos
- 1.3. Programación imperativa o por procedimientos
- 1.4. Programación funcional
- 1.5. Lenguaje estático y Tipado estático
- 1.6. Lenguaje dinámico y Tipado dinámico

2. Organización de un programa informático avanzado

- 2.1. Principales características
- 2.2. Paradigmas de diseños
- 2.3. Testing, calidad y despliegues

3. Entornos

- 3.1. Nativo
- 3.2. Navegador
- 3.3. Servidor
- 3.4. Runtime (entorno de ejecución)

4. Programación Orientada a Objetos

- 4.1. Clases
- 4.2. Objetos
- 4.3. Interfaces
- 4.4. Herencia

5. Algorítmica

- 5.1. Algoritmos de ordenación
- 5.2. Algoritmos de búsqueda
- 5.3. Algoritmos de filtrado
- 5.4. Algoritmos avanzados



6. Debugging y Refactoring

- 6.1. Herramienta para debugging (ejemplo con VS)
- 6.2. Testeo básico en javascript

7. Programación Funcional: Visión

- 7.1. Funciones
- 7.2. Paso de parámetros por valor.
- 7.3. Paso de parámetros por referencia
- 7.4. Estructuras de datos
- 7.5. Funciones de Orden Superior

8. Planificación de proyectos software

- 8.1. Cómo desarrollar un proyecto
- 8.2. Ciclo de vida del software
- 8.3. Paradigmas de desarrollo de software
- 8.4. Fases del ciclo de vida
- 8.5. Análisis de casos reales



Módulo 5. Introducción a BBDD

1. Introducción a las Bases de Datos

- 1.1. Configuración del entorno de trabajo

2. Tipos de BBDD

- 2.1. Sistemas transaccionales y analíticos
- 2.2. Sistemas relacionales y documentales

3. Sistemas Gestores de BBDD

- 3.1. MySQL. Historia y características

4. Introducción al uso de IDE para trabajar con MySQL

- 4.1. Configuración de MySQL Workbench
- 4.2. MySQL Workbench para tareas de desarrollo y modelado de datos

5. Fundamentos de SQL

- 5.1. Creación de tablas
- 5.2. Definiendo la integridad de Datos
- 5.3. Tipos de datos.
- 5.4. Valores NULL y Three-ValuedLogic

6. Técnicas para la modificación de datos

- 6.1. INSERT
- 6.2. UPDATE
- 6.3. DELETE

7. Técnicas para la obtención de datos

- 7.1. SELECT, WHERE
- 7.2. DISTINCT, GROUP BY, HAVING
- 7.3. Vistas. Propósito y ejemplos de uso.



8. Generación y uso de secuencias

- 8.1. Definición de una secuencia
- 8.2. Obteniendo valores de secuencia
- 8.3. Reenumeración de secuencias

Módulo 6. Bases de Datos avanzadas

¿Qué vas a conseguir en este módulo?

Al final de esta materia el alumno conocerá conceptos avanzados de Bases de Datos. Aprenderá aspectos más profundos del lenguaje SQL, aprenderá a desarrollar consultas más complejas que forman parte del uso cotidiano de este lenguaje, además de trabajar con rutinas almacenadas como: procedimientos almacenados, funciones y disparadores. Adquirirá los conocimientos necesarios para medir el rendimiento de una consulta y aprenderá a aplicar técnicas de optimización, así como el diseño apropiado de estrategias de indexación. También será capaz de reconocer e implementar buenas prácticas de programación.

1. Consultas avanzadas

- 1.1. Subconsultas
- 1.2. Tipos de JOIN

2. Trabajando con cadenas

- 2.1. Propiedades de cadena
- 2.2. Funciones de cadena
- 2.3. Controlando la sensibilidad en comparaciones de cadena
- 2.4. Patrones de coincidencia con SQL

3. Gestión de duplicados

- 3.1. Introducción
- 3.2. Prevención de duplicidad en tablas
- 3.3. Contabilizando e identificando duplicados
- 3.4. Eliminación de duplicados en tablas

4. Transacciones

- 4.1. Definición
- 4.2. START TRANSACTION y COMMIT/ROLLBACK
- 4.3. Niveles de aislamiento

5. Procedimientos almacenados

- 5.1. Creación y mantenimiento
- 5.2. Cursores y control de errores



6. Funciones

- 6.1. Integradas en el SGBD y definidas por usuario
- 6.2. Funciones de agregación y escalares

7. Índices

- 7.1. Tipos de índices
- 7.2. Estrategias de indexación para un alto rendimiento
- 7.3. Mantenimiento de índices

8. Estrategias de optimización

- 8.1. Análisis de planes de ejecución
- 8.2. Recopilando estadísticas de ejecución
- 8.3. Aplicación de técnicas de optimización

Módulo 7. Git & GitHub: Avanzado

1. Git

- 1.0. Uso avanzado de la consola
- 1.1. Rama o branch
- 1.2. Mezcla o merge. Tipos
 - 1.2.1. Fast forward
 - 1.2.2. Threeway
 - 1.2.3. Rebase
- 1.3. Archivos ignorados
- 1.4. Stash
- 1.5. Protocolos: Local, HTTPS, SSH y GIT

2. GitHub

- 2.1. Administración básica
- 2.2. Fork

3. GitFlow



Módulo 8. Javascript avanzado & EcmaScript6+

1. Curiosidades del texto

1.1 Caracteres y secuencias de escape

2. Propiedades de datos y propiedades de acceso

3. Alcance -Scope y Closure

- 3.1. Motivación
- 3.2. Alcance Local
- 3.3. Alcance Global
- 3.4. Resolución de Alcance
- 3.5. Closures
- 3.6. Espacios de nombres (Namespaces)
 - 3.6.1. Duplicados de nombres

4. Colecciones y conjuntos

- 4.1. Arrays en JavaScript
 - 4.1.1. Construcción y acceso de elementos
 - 4.1.2. Asociativos
 - 4.1.3. Tipados
- 4.2. DataView, Set y Map
- 4.3. Generadores, iteradores, iterables y for... of

5. ES6

- 5.1. Constantes
- 5.2. Tipos de dato
- 5.3. Template Literals
- 5.4. Mejoras en las funciones
- 5.5. Funciones flecha
- 5.6. Destructuración
- 5.7. Sets y Maps
- 5.8. Symbols
- 5.9. Promises



Módulo 9. JQuery

1. Introducción a jQuery

- 1.1 jQuery y JavaScript
- 1.2 Obteniendo / Accediendo a la biblioteca de jQuery

2. El modelo de eventos jQuery

- 2.1 Eventos en Javascript y JQuery.
- 2.2 Manejadores de Eventos.
- 2.3 Métodos de manipulación de eventos

3 Selectores jQuery

- 3.1 Selectores de posición
- 3.2 Selectores jQuery personalizados

4 Ajustando objetos DOM con jQuery

- 4.1 Obtener objetos por ID, Clase, Etiqueta
- 4.2 Manejo de eventos con jQuery

5 jQuery y llamadas AJAX

- 5.1 Usando la API Ajax
- 5.2 Cargando datos con GET & POST
- 5.3 Trabajando con datos JSON
- 5.4 Serializando su manejo de formulario con serialize
- 5.5 Manejo de una solicitud AJAX completa
- 5.6 Consideraciones de Seguridad



Módulo 10. Node.JS & Express.JS

1. Introducción

- 1.1. ¿Qué es NodeJS?
- 1.2. ¿Por qué NodeJS?
- 1.3. EventLoop

2. Core de NodeJS

- 2.1. Global
- 2.2. Process
- 2.2. Http
- 2.3. Buffer
- 2.4. Streams
- 2.5. Events
- 2.6. Util
- 2.7. Timers
- 2.8. modules

3. Core de NodeJS (II)

- 3.1. Http: Event, streams, fs
- 3.2. Path
- 3.3. URL
- 3.4. QueryString
- 3.5. Static files
- 3.6. Router
- 3.7. HTTPclient
- 3.8. GET/POST



4. Desarrollo de API usando Express

- 4.1. Express básico: hello, file, request, response
- 4.2. Uso de middlewares
- 4.3. Organización de código: express-generator
- 4.4. API RESTful: Implementación de CRUD de películas (RAM)
- 4.5. Persistencia de datos: MySQL
- 4.6. Persistencia de datos: MongoDB
- 4.7. Persistencia de datos: ORM (TypeORM).

5. Desarrollo de API/Microservicio usando boilerplate ES6-Express (Microservicio1) conectado a MongoDB

6. Desarrollo de API/Microservicio usando boilerplate TypeScript (Microservicio2) conectado a MySQL

7. Desarrollo de API/Microservicio usando NestJS (Microservicio3) conectado a PostgreSQL

8. Testing en Node

- 8.1. Pruebas unitarias en node - KATA 1. (mocha)
- 8.2. Pruebas unitarias en node - KATA 2. (mocha)
- 8.3. Pruebas unitarias de servicios usando jest (NestJS)
- 8.4. Pruebas unitarias de controllers usando jest (NestJS)
- 8.5. Pruebas de integración de servicios/controllers usando jest (NestJS)
- 8.6. Pruebas e2e usando supertest y jest (NestJS)



Módulo 11. Introducción a Docker

1.1. Qué es Docker

1.2 Para qué sirve Docker

1.3 Instalación

1.4 Docker Images

1.4 Docker Containers

1.4 Docker Volumes



Módulo 12. MongoDB

1. Introducción a MongoDB

- 1.1 Descripción general de MongoDB
- 1.2 MongoDB: fundamentos
- 1.3 Arquitectura de MongoDB
- 1.4 Instalando MongoDB en local
- 1.5 Servicios Cloud para MongoDB

2 Modelado de datos con MongoDB

- 2.1 Modelado de datos orientados a documentos
- 2.2 Modelado de datos en RDBM versus MongoDB
- 2.3 Optimización y uso de índices

3 Sintaxis de consulta MongoDB

- 3.1 El caparazón de mongo
- 3.2 Operaciones CRUD
- 3.3 Consultas de agregación
- 3.4 Optimización y mejores prácticas

4 Mongoose

- 4.1 Inicialización del proyecto
- 4.2 Esquema y modelo
- 4.3 El método find
- 4.4 Tipos de schema
- 4.5 Setter y Getters
- 4.6 Hooks

5 CRUD con MongoDB, NodeJS y Express

- 5.1 Ejemplo práctico de aplicación CRUD con MongoDB, NodeJS y Express



Módulo 13. Diseño de APIS.REST

1. Introducción

- 1.1. De dónde partimos
- 1.2. Servicios Web
- 1.3. Ejemplos

2. ¿Qué es REST?

- 2.1. Definición
- 2.2. API REST
- 2.3. Restful
- 2.4. Ejemplos

3. Arquitectura REST

- 3.1. Interfaz Uniforme
- 3.2. Sin estado
- 3.3. Separación entre cliente y servidor

4. URIs

- 4.1. Estructura
- 4.2. Sustantivos no verbos
- 4.3. Buenas prácticas
- 4.4. Versionado
- 4.5. Ejemplos

5. HTTP

- 5.1. Métodos
- 5.2. Códigos de estado
- 5.3. Ejemplos



6. Autenticación

- 6.1. Tokens
- 6.2. OAuth
- 6.3. Ejemplos

7. Ejercicio Final



Módulo 14. React & Hooks

1 React

2 Introducción a React

- 2.1 ¿Qué es un Framework?
- 2.2 ¿Qué es React?
- 2.3 Principios y ventajas de React
- 2.4 El Virtual DOM
- 2.5 ¿Qué es Babel?
- 2.6 Herramientas útiles
- 2.7 Instalación y hola mundo
- 2.8 Lenguaje JSX

3 Primer proyecto en React

- 3.1 Estructura de Componentes y Encapsulación
- 3.2 Propiedades y comunicación de componentes
- 3.3 Primera refactorización (export/import)
- 3.4 Creando un header y footer
- 3.5 Creando un layout
- 3.6 Aplicando estilos
- 3.7 Eventos sintéticos

4 Interactividad en React

- 4.1 El estado del componente
- 4.2 Actualizando nuestro estado
- 4.3 La actualización del componente y re-render
- 4.4 Bucles: pintando una lista
- 4.5 La propiedad Key

5 Fetch y llamadas HTTP con React

- 5.1 Actualizando el estado con Fetch
- 5.2 Prueba técnica del DOM Virtual



6 Comunicación de alto nivel

- 6.1 Creando nuestra página de Login
- 6.2 Abstracción de bajo nivel
- 6.3 Condicionales en React
- 6.4 Subir eventos a través de props

7 Tipos de componentes

- 7.1 Statefull
- 7.2 Stateless
- 7.3 PureComponent

8 Extras

- 8.1 ReactDevelopertools
- 8.2 PropTypes
- 8.3 DefaultProps
- 8.4 ModuleStyles
- 8.5 CssModules
- 8.6 Decoradores

9 ReactRouter

10 Introducción a ReactRouter

- 10.1 ¿Qué es ReactRouter?
- 10.2 Tipos de Router
- 10.3 Manejando las Rutas
- 10.4 Link y NavLink
- 10.5 Parámetros de ruta

11 ReactRouter Avanzado

- 11.1 Redirecciones
- 11.2 Rutas privadas
- 11.3 Páginas por defecto 404
- 11.4 Configuración del Router



12 Arquitectura de una tienda online

- 12.1.1 Inicialización del entorno
- 12.1.2 Configuración del Module Bundling
- 12.1.3 Preparación del linter
- 12.1.4 Creación del router

13 Inicio del proyecto

- 13.1.1 Creación de la pantalla de Login
- 13.1.2 Home de la tienda
- 13.1.3 Categorías
- 13.1.4 Carrito de la compra
- 13.1.5 Proceso de pago

14 Comunicación con el servidor

- 14.1.1 Lectura de la documentación
- 14.1.2 Validación del login
- 14.1.3 Añadir Fetch en los productos
- 14.1.4 Enviar Compra y confirmación

15 Hooks

- 15.1 Introducción a Hooks
- 15.2 Inicio de un proyecto con Hooks
- 15.3 useState
- 15.4 useCounter - CustomHook
- 15.5 useEffect - SimpleForm
- 15.6 useEffect unmount - Cleanup
- 15.7 useEffect - Precauciones
- 15.8 Formulario con custom Hook
- 15.9 useFetch - CustomHook
- 15.10 useMemo
- 15.11 useCallback



Módulo 15. Full Stack: de 0 a producción

1. Introducción

- 1.1 Concepto de idea
- 1.2 Brainstorming
- 1.3 Ejemplos

2. Diseño de producto

- 2.1 Introducción
- 2.2 Mapa de Empatía
- 2.3 EDV Framework (Entender, Definir y Validar)
- 2.4 Tamaño del Mercado
- 2.5 Curva de valor
- 2.6 Business ModelCanvas
- 2.7 Ejemplos

3. Prototipado

- 3.1 Tipos
- 3.2 MVP (Mínimo Producto Viable)
- 3.3 Herramientas de prototipado
- 3.4 Ejemplos

4. Empezar un proyecto

- 4.1 Elegir un lenguaje de programación
- 4.2 Editores
- 4.3 Welcomepack
- 4.4 Documentación de código
- 4.5 Entornos (Desarrollo, Preproducción y Producción)
- 4.6 Ejemplos

5. Desarrollo

- 5.1 Repositorios de código
- 5.2 Check Style y normas de CleanCode
- 5.3 Ejemplos



6. Entregas

- 6.1 Integración Continua (Modelo y Herramientas)
- 6.2 Pruebas
- 6.3 Crear ramas en el repositorio
- 6.4 Versionar
- 6.5 Resolución de incidencias
- 6.6 Documentación de las entregas
- 6.7 Ejemplos

7. Producción

- 7.1 Cierre del desarrollo, versión final
- 7.2 Manual de la aplicación
- 7.3 Equipo de mantenimiento
- 7.4 Propuestas de valor
- 7.5 Ejemplos

8. Proyecto-Ejercicio Final



Módulo 16. Gestión de proyectos: Scrum

1. Introducción

- 1.1. ¿Qué es la gestión de proyectos?
- 1.2. Gestión tradicional o en cascada
- 1.3. Construir software no es construir un puente.
- 1.4. Ejemplos

2. Lo tradicional

- 2.1. Toma de requisitos
- 2.2. Roles
- 2.3. Planificación
- 2.4. Herramientas y diagramas (casos de uso, diagramas de flujo...)

3. Gestión ágil

- 3.1. Manifiesto de agilidad
- 3.2. Objetivos de una metodología ágil

4. Kanban

- 4.1. Introducción
- 4.2. WIP
- 4.3. Equipo y proyecto
- 4.4. Herramientas (Pizarra)
- 4.5. Ejemplos

5. XP Programming

- 5.1. Introducción
- 5.2. Metodología



6. Introducción a Scrum

- 6.1. Principios básicos
- 6.2. Scrum vs Tradicional

7. Scrum: Roles y Artefactos

- 7.1. Roles
- 7.2. Artefactos
- 7.3. Ejemplos

8. Scrum: Ciclo

- 8.1. Esquema general
- 8.2. Product Backlog
- 8.3. Historias de usuario
- 8.4. Estimación
- 8.5. Sprint Planning
- 8.6. Sprint Backlog
- 8.7. Sprint
- 8.8. Daily
- 8.9. Sprint Review y Restropective

9. Scrum: Buenas prácticas

- 9.1. Falso Scrum
- 9.2. Recomendaciones

10. Proyecto



Módulo 17. Calidad de Software: QA

1. Introducción
2. Conceptos de Calidad del Software
3. Habilidades de un QA
4. Diseño y ejecución de pruebas manuales
5. Reporte de fallos
6. Pruebas automáticas de UI con Cypress
7. Pruebas de API con Postman
8. Pruebas Unitarias con Jest