= Temario de la mentoria en PROGRAMACION

La mayoria de temas relevantes a ver son presentados.

Nota: Si bien los temas generales estan en orden (adaptacion, ciclo basico, paradigmas, etc), los subtemas NO estan ordenados (impreviscible el contexto del alumno)

Nota 2: Si, son varios temas. Pero empezaremos tan facil como te sea necesario. Al final, es para vos.

Nota 3: Tenes material gratuito de Jose en youtube y Linkedin presentando varios de estos temas (aunque la minoria) (enlaces: youtube = https://www.youtube.com/channel/UCNv4zj_4zv41LNpysCJdaFQ linkedin = https://www.linkedin.com/in/jose-pepe-web-soft-dev/)

== Adaptacion

. Perifericos (si es necesario) . Herramientas de edicion de texto . Emuladores de terminales . Sistema operativo Windows/Linux

== Ciclo Basico

. Versionado con git . Funcionamiento local . Interaccion con repositorios remotos . Interaccion con codigo remoto . Commits convencionales . Logica de programacion imperativa . Logica de programacion declarativa . Hubs de repositorios . Github Actions . Crear una cuenta . Crear repositorios . Manejar configuracion basica . Dato vs informacion . Logica basica

== Paradigmas de interes para el backend

. Imperativo: . procedural . Orientacion a objetos . Declarativo: . Funcional . Reactivo . Implementaciones declarativas en programacion imperativa . Ejemplos en Python y Java . Lambdas . Streams . Filter . Maps . Reducers

== Informatica basica

. A que herramientas estamos programando . Cpu . Placa madre . Memorias (principal, secundaria, cpu cachees, placa madre cachees, ROM, swap) . buses . Por que usar lenguaje para programar . Archivo, diferentes maneras de persistencia . Bases de datos

== Programacion basica

. Criterios para eleccion de paradigmas . Librerias y frameworks . Componentes vs Modulos . Flujos de informacion, pipelines . Estructuras de decision . Llamadas y procedimientos . Declaracion y asignacion . Entrada y salida de datos estandars . Que es un pipeline . Introduccion a los Principios SOLID . Interfaces (o protocolos) . Refactorizacion

== Arquitectura basica

. Principio de responsabilidad unica . Principio de modularidad . Cohesion . Acoplamiento . Relacion cohesion-acoplamiento . Niveles de cohesion-acoplamiento . Principio del desarrollo iterativo incremental . Principio del retraso de la decision segun cono de incertidumbre . Principio de la ocultacion de informacion . Data transfer objects . Encapsulacion vs ocultacion . Consecuencias de los principios . Reutilizacion de codigo y menos duplicidad . Mayor interdependencia . Mayor control del producto . Capas de abstraccion . Diseño arquitectonico basico . Por capas . Introduccion a eventos . Introduccion a Domain Driven Design . Testing . Tests unitarios . Tests de integracion

== Introduccion al agilismo

. Propiedades y beneficios del agilismo . Contras del agilismo . Definicion de 'Valor para el usuario' . Ley de 'antes mejor' . Fallar ASAP . Entregar ASAP . Comunicar ASAP . Alternativa: cascada

== Programacion intermedia

. Profundizacion en SOLID . Clean Code . Introduccion a Python . TDD: . Flujo de Trabajo . Red, Green, Blue . De casos particulares a abstraccion, clases y superclases . Interaccion con el versionado . Variaciones: Tests de integracion antes de los unitarios . Documentacion ejecutable . Introduccion a los Patrones de Diseño . Tipo de patrones . Señales de TDD para implementarlos . Tipos de objetos . Encapsulamiento: . La logica de losgetters y setters . Provider vs Consumer . Scripting

== Arquitectura intermedia

. Arquitectura modular . Arquitectura evolutiva . Modelado arquitectonico . DDD . Documentacion: . C4 . UML . Template Arch42 . Estilos de arquitectura: . MVC . REST . Introduccion a los patrones arquitectonicos . Atributos de calidad . Fitness Functions . Acoplamiento/cohesion por cada patron

== Containerizacion con Docker

. Containerizar . Docker . Imagenes . Hub . Contenedores . Docker cli . Docker commits

El temario sigue, y los puntos tratados aqui se presentan con la profundidad necesaria, y en el orden apropiado, segun cada estudiante. Es por la relacion intensa entre todos los puntos que los temas son ciclicos: Si bien el progreso mantiene el orden general aqui presentado, los temas anteriores se retomaran a medida que el avance requiera profundizarlos.