

BLOC A - JHON

- Característiques del desenvolupament web: web development stack, backend, frontend, API REST, peticions http, serveis,
- Spring framework: característiques, utilitats, components, versions, entorns de desenvolupament adients

punt 1

(par1)

Un desenvolupador web [FULL STACK](#) és una persona que pot desenvolupar programari tant de client com de servidor.

Per un altre costat, un desenvolupador Full Stack és l'encarregat de manejar cada un dels aspectes relacionats amb la creació i el manteniment d'una aplicació web. Per a ell és fonamental que el desenvolupador Full Stack tingui coneixements en desenvolupament Front-End i Back-End a més de manejar diferents sistemes operatius i llenguatges de programació.

A més de dominar [HTML](#) i [CSS](#), també sap com:

Programa un navegador (utilitzar [JavaScript](#), [jQuery](#), [Angular](#) o [Vue](#))

Programar un servidor (PHP, [ASP](#), Python o Node)

Programa una base de dades ([SQL](#), [SQLite](#) o [MongoDB](#))

Avantatges

L'avantatge de ser un desenvolupador web full stack és:

- Dominar totes les tècniques implicades en un projecte de desenvolupament
- Fer un prototip molt ràpidament
- Proporcionar ajuda a tots els membres de l'equip
- Reduir el cost del projecte
- Reduir el temps utilitzat per a la comunicació en equip
- Canviar entre el desenvolupament frontal i posterior segons els requisits
- Entendre millor tots els aspectes de les noves i properes tecnologies

Desavantatges

- La solució escollida pot ser errònia per al projecte
- La solució escollida pot dependre de les habilitats dels desenvolupadors

- La solució pot generar un risc per a les persones clau
- Ser desenvolupador full stack és cada vegada més complex

(par2)

BACKEND es la part del desenvolupament que s'encarrega de la lògica de una web o aplicació, també és la capa d'accés a dades d'un programari o qualsevol dispositiu, que no és directament accessible pels usuaris.

Alguns dels llenguatges de programació per Backend són PYTHON , NODE.JS , PHP , GO , RUBY i C # . I així com en el frontal, tots aquests llenguatges tenen diferents frameworks que et permeten treballar millor segons el projecte que estàs desenvolupant

Funcions backend

- desenvolupament de funcions que simplifiquen el procés
- lògica
- connexió amb base de dades
- ús de llibreries del servidor
- coneixements de frameworks
- tipus de BD

(part3)

FRONTEND a diferència del backend es la part visual de la pàgina web, llavors es la que interactua directament amb l'usuari. Són totes les tecnologies de disseny i desenvolupament web que corren en el navegador i on definir coses com:

- tipus de lletra
- colors
- adaptació de pantallas
- moviments
- efectes visuals

(part4)

(Les API són conjunts de definicions i protocols que s'utilitzen per dissenyar i integrar el programari de les aplicacions. Suele considerar-se com el contracte entre el proveïdor d'informació i l'usuari, on s'estableix el contingut que es necessita per part de l'consumidor (la trucada) i el que requereix el productor (la resposta.))

resum: En altres paraules, les API li permeten interactuar amb un ordinador o un sistema per obtenir dades o executar una funció, de manera que el sistema

comprengui la sol·licitud i la compleixi.)

Una [API REST](#), o API de RESTful, és una interfície de programació d'aplicacions (API o API web) que s'ajusta als límits de l'arquitectura REST i permet la interacció amb els serveis web de RESTful. Es recolza totalment en l'estàndard HTTP . Va ser definida el 2000 per [Roy Fielding](#).

Ara per ara, la majoria de les aplicacions que es desenvolupen per a serveis professionals disposen d'una API REST per a l'intercanvi d'informació entre el front i el back . El que la fa tan potent és precisament l'aïllament que proporciona entre la lògica del back-end i qualsevol client consumidor d'aquest. Això li permet ser usada per qualsevol tipus de client: web, mòbil, etc. Així, qualsevol dispositiu / client que entengui d'HTTP pot fer ús de la seva pròpia API REST de manera molt simple.

(part5)

Els missatges [HTTP](#), són els mitjans pels quals s'intercanvien dades entre servidors i clients. Hi ha dos tipus de missatges: peticions , enviades pel client a servidor, per demanar l'inici d'una acció; i respostes , que són la resposta de servidor.

Les peticions HTTP són missatges enviats per un client, per iniciar una acció al servidor.

La seva línia d'inici està formada per tres elements:

- Un [mètode HTTP](#) , un verb com: [GET](#), [PUT](#) o [POST](#) o un nom com: [HEAD](#) o [OPTIONS](#), que descriuen l'acció que es demana sigui realitzada.
- L'objectiu d'una petició, normalment és una [URL](#), o l'adreça completa del protocol, port i domini també solen ser especificats pel context de la petició.
- La versió d'HTTP, la qual defineix l'estructura dels missatges, actuant com a indicador, de la versió que espera que es faci servir per a la resposta.

(part 6)

Els [SERVEIS WEB](#) són aplicacions modulars autocontenides que pot descriure, publicar, localitzar i invocar a través d'una xarxa, llavors es pot dir que realitza un conjunt específic de tasques.

També podem dir que es tracta d'un entorn estandarditzat per distribuir la comunicació entre aplicacions client i servidor a la World Wide Web . Pot editar-se i comptar amb paquets de millora.

punt 2

(part1)

Un framework és un marc o esquema de treball utilitzat per agilitzar els processos de desenvolupament ja que evita haver d'escriure codi de forma repetitiva, assegura unes bones pràctiques i la consistència de el codi.

Quan parlem de [SPRING](#) és un framework de el llenguatge de programació Java que ens permet desenvolupar aplicacions de manera més ràpida, eficaç i curta, saltant-nos tasques repetitives i estalviant-nos línies de codi.

(part2)

L'ús de Spring té nombrosos avantatges, entre elles:

- Injecció de dependència (DI) : tècnica o patró de disseny utilitzat com una de les formes d'inversió de control (IOC) per afavorir el baix acoblament. És a dir, quan una classe necessita d'una altra classe (dependència), en lloc de crear l'objecte dins de la pròpia classe, Spring s'ho subministra o s'injecta. Aquesta forma de IOC segueix el principi de Hollywood (no ens truquis, nosaltres et telefonem) en què l'aplicació és la que pren el control en lloc de el codi, aportant els següents beneficis:

- Proporciona modularitat.
- Escalabilitat de l'aplicació sense necessitat de modificar les classes.
- Evita la dependència entre classes

- Desenvolupament senzill amb [POJOs](#) (Plain Old Java Objects)

- Minimitza el codi repetitiu

- Simplifica l'accés a dades gràcies als [ORM](#) (Object Relational Mapping)

- Programació Orientada a Aspectes ([AOP](#)): Permet una modularització molt més alta per fer una separació molt clara de les diferents tasques que ha de realitzar cada classe en la nostra aplicació.

(part3)

Spring Framework comprèn diversos mòduls que proveeixen un rang de serveis:

- **Contenedor d'inversió de control** : permet la configuració dels components d'aplicació i l'administració de el cicle de vida dels objectes Java, es porta a terme principalment a través de la injecció de

dependències .

- **Programació orientada a aspectes** : habilita la implementació de rutines transversals.
- **Accés a dades** : es treballa amb [RDBMS](#) a la plataforma Java, usant Java Database Connectivity i eines de Mapeig objecte relacional amb bases de dades [NoSQL](#) .
- **Gestió de transaccions** : unifica diferents APIs de gestió i coordina les transaccions per als objectes Java.
- **Model vista controlador** : Un framework basat en HTTP i servlets , que proveeix eines per a l'extensió i personalització d'aplicacions web i serveis web REST .
- **Framework d'accés remot** : Permet la importació i exportació estil [RPC](#) , d'objectes Java a través de xarxes que suporten [RMI](#) , [CORBA](#) i protocols basats en HTTP incloent serveis web ([SOAP](#)).
- **Convenció sobre Configuració** : el mòdul Spring Roo ofereix una solució ràpida per al desenvolupament d'aplicacions basades en Spring Framework, privilegiat la simplicitat sense perdre flexibilitat.
- **Processament per lots** : un framework per a processament de molt volum que com a característiques inclou funcions de registre / traçat, maneig de transaccions, estadístiques de processament de tasques, reinici de tasques, i maneig de recursos.
- **Autenticació i Autorització** : processos de seguretat configurables que suporten un rang d'estàndards, protocols, eines i pràctiques a través del subprojecte Spring Security (antigament Acegi).
- **Administració Remota** : Configuració de visibilitat i gestió d'objectes Java per a la configuració local o remota via [JMX](#) .
- **Missatges** : Registre configurable d'objectes receptors de missatges, per al consum transparent des de la a través d' JMS , una millora de l'enviament de missatges sobre les API JMS estàndard.
- **Testing** : Suport de classes per a desenvolupament d'unitats de prova i integració.

(part4)

Els primers components del que s'ha convertit en Spring Framework van ser escrits per [Rod Johnson](#) en l'any 2000.

Versions

0.9 (2002)	1.0 (2004)	1.2.4 (2004)	2.0 (2006)
2.5 (2007)	3.0 (2009)	3.1 (2011)	3.2 (2013)
4.0 (2017)	4.3.7 (2017)	5.0 (2017)	

(part5)

- Java
- Kotlin
- Groovy

