



Planear actividades de construcción del software de acuerdo con el diseño establecido.

Producción y compilación del contenido digital Inst. Angélica M. Triana
Solo fines académicos

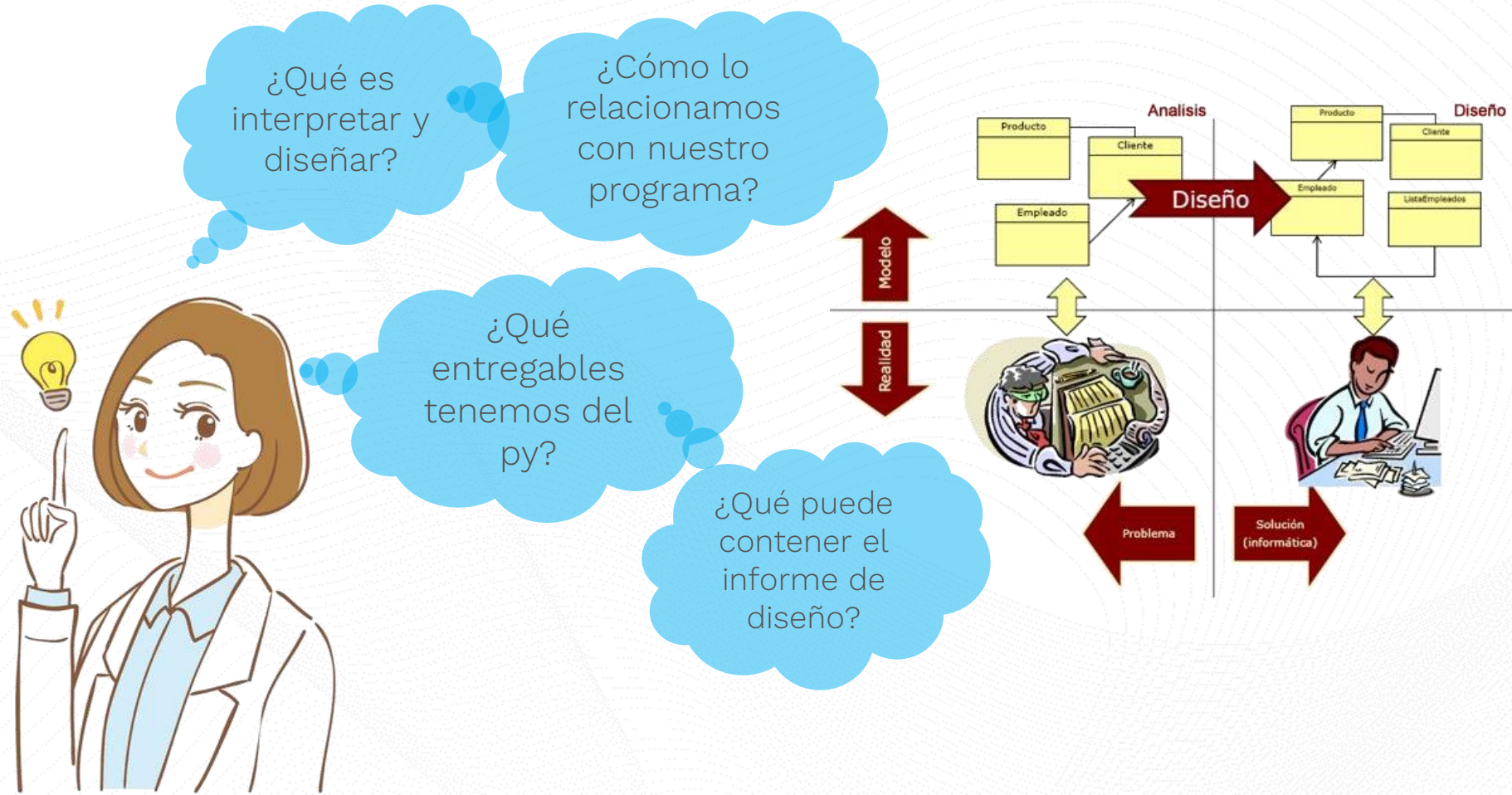


www.sena.edu.co

Contenido

Revisión entregables proyecto
Herramientas de desarrollo
Proceso para la instalación y
configuración de plataformas de
desarrollo.
Actividades de la sesión

Reflexión inicial



Actividad de la sesión



Actividad de la sesión

Por grupo de proyecto, realizar un informe a través de <https://fliphtml5.com/es/> que incluya lo siguiente:

- Nombre del proyecto e integrantes del grupo
- Objetivo general y alcance del proyecto
- Metodología de desarrollo (si aplica)
- Determinar los requisitos funcionales y no funcionales del software de acuerdo con los requerimientos del cliente. (Mostrar solo la lista de req.)
- Determinar las especificaciones técnicas (hw y sw) a partir del diagnóstico de necesidades informáticas (si aplica)
- Diagramas UML relacionado con el proyecto. (Los generados hasta el momento)
- MER (si aplica)

Revisar si el proyecto cumple con lo que se requiere en el tecnólogo: un software orientado a la web que haga entrada-procesamiento de datos (información)-salida-reportes.

Finalmente, comparta el link en el espacio indicado por el instructor.

Reflexión inicial



¿Se han planteado los requerimientos de hardware?

¿Se ha utilizado herramientas para el seguimiento de los entregables del proyecto?



¿Qué herramientas podemos utilizar para el proceso de desarrollo de software?



Hardware necesario para mi sistema

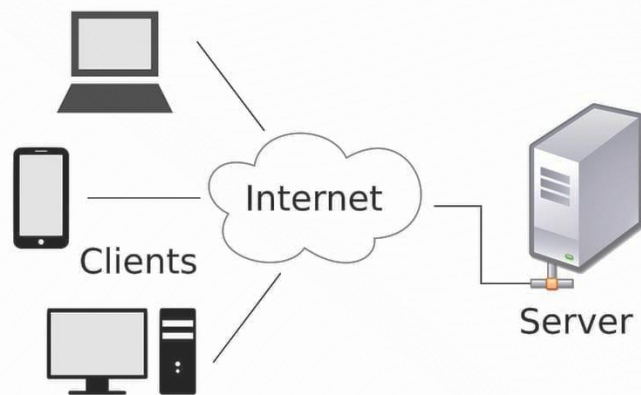


Requisitos del Hardware:

Son las características mínimas que debe tener el hardware de una computadora para poder soportar y/o ejecutar una aplicación o un dispositivo específico.

Recuperado de

<https://clic.unab.edu.co/index.php?/Knowledgebase/Article/View/48>



Se debe tener en cuenta el servidor.

Servidor web



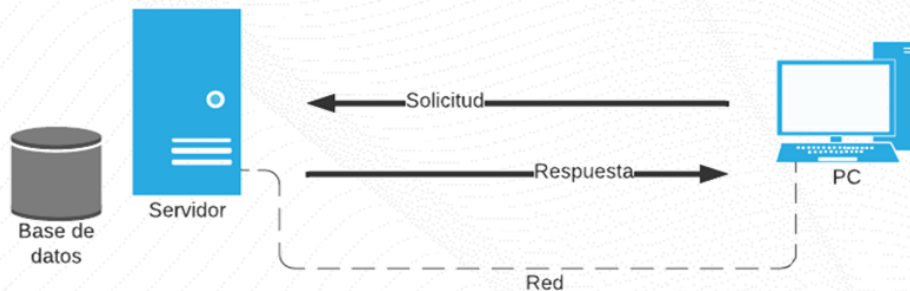
Puede referirse a dos cosas: al software del servidor (es decir, al programa en sí) y al equipo en el que el software del servidor web se está ejecutando.

Cuando se trata de este último, se habla del host (un host puede alojar múltiples soluciones de software para servidores web).

Recuperado de

<https://www.ionos.es/digitalguide/servidores/know-how/servidor-web-definicion-historia-y-programas/>

Modelo cliente-servidor



Diseño de la Arquitectura DEL SOFTWARE

La estructura del sistema informático compuesto por subsistemas, componentes, interfaces y las relaciones entre ellos.

es

Ejemplos

Estructura empresarial en capas

Modelos

Dos capas

Tres capas

Multicapa

Elementos

Capa de datos

Capa de reglas de negocio

Capa de presentación

Diagrama de paquetes

Elementos

Paquetes

Dependencias

Representación gráfica

Diagrama de Componentes

Elementos

Componentes

Puertos

Interfaces

permite modelamiento de

Estructuras

de tipo

Modulares

Por componente y conector

De asignación

las cuales pueden ser

Estructura de composición y descomposición

Tipo "uses"

De clase

En capas

las cuales pueden ser

De proceso

De concurrencia

Cliente-servidor

Repositorio

las cuales pueden ser

De despliegue

De asignación de trabajo

De implementación

beneficios de su definición

1. Mecanismo de comunicación entre stakeholders

2. La arquitectura focaliza la implementación.

3. El desarrollo del proyecto se organiza sobre los elementos arquitectónicos

4. La arquitectura permite o inhabilita la consecución de atributos de calidad

5. La arquitectura es la clave para manejar el cambio

6. La arquitectura es la base para el desarrollo incremental.

7. La arquitectura es un modelo que permite la reutilización.

Front-end y Back-end



Reflexión inicial



¿En qué consiste el front-end y el back-end?

¿Conoce algunas herramientas relacionadas con el front y el back?

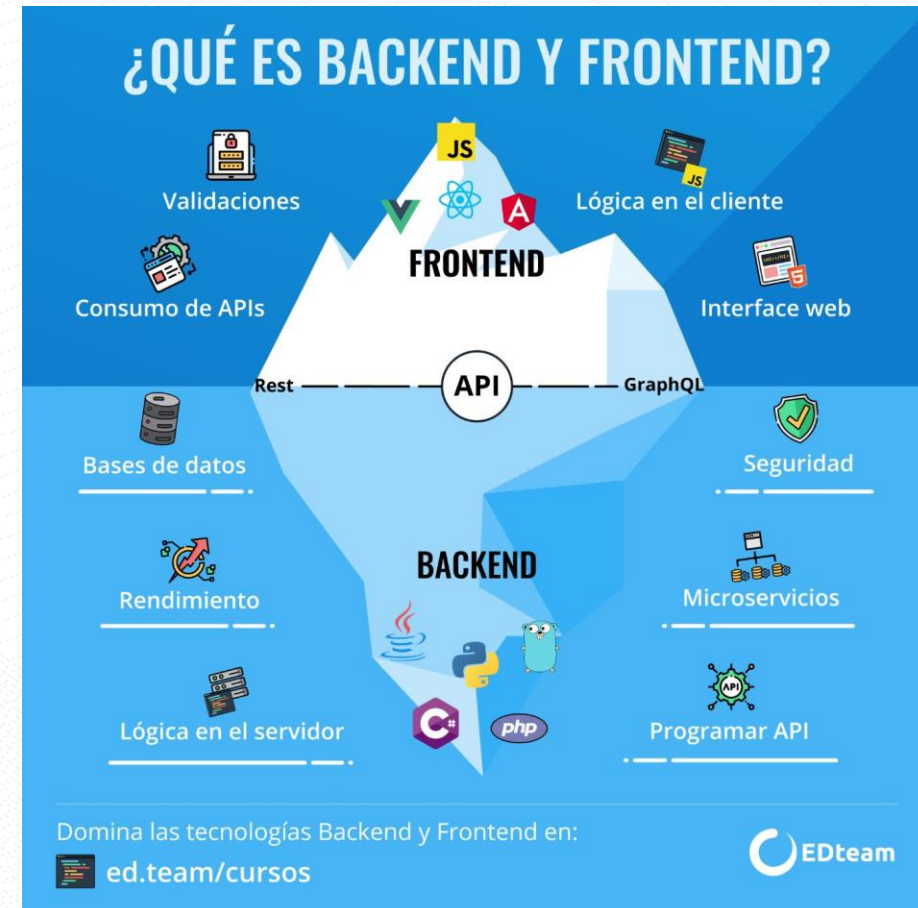
¿Cómo relacionamos el informe de diseño con las herr. de desarrollo y seguimiento?

Herramientas de desarrollo de software (parte 1)



Dentro del área de programación web, existe 3 tipos de programador. Cada uno posee sus características, también ventajas y desventajas; además de seguir un lenguaje específico. Algunos factores deben evaluarse con cautela.

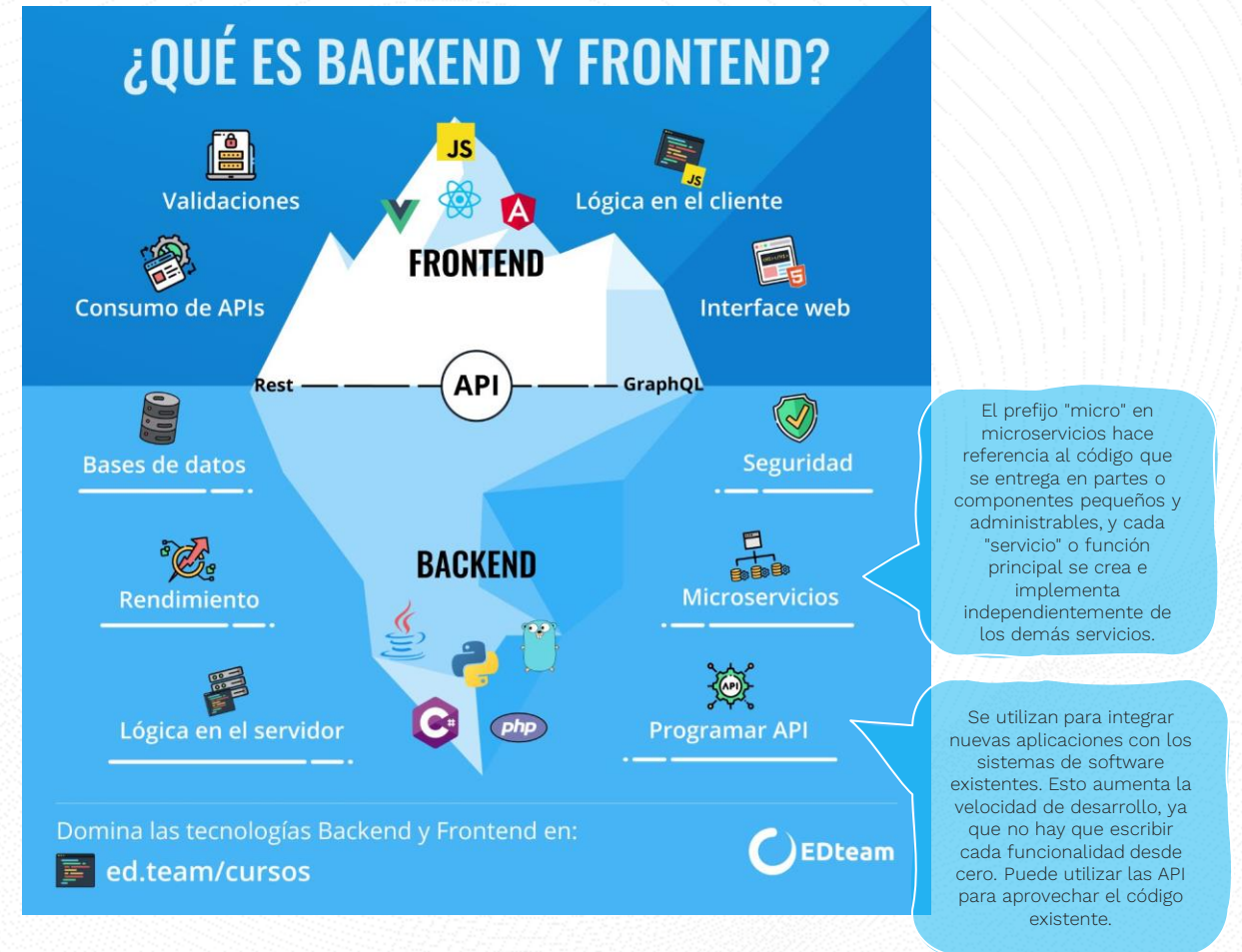
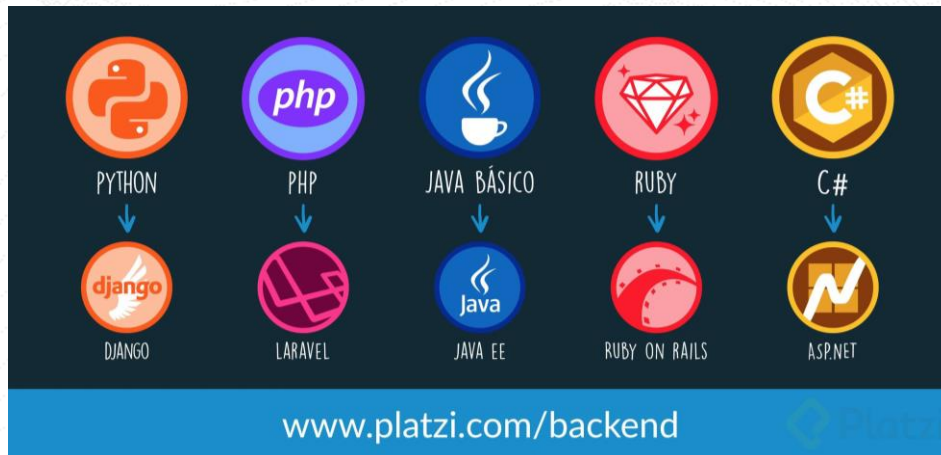
Al optar por **Front-End**, es necesario saber desarrollar pantallas de aplicación proyectadas por un arquitecto o diseñador web. Para esto, generalmente se necesita tener conocimiento en **HTML, CSS y JavaScript**.



Herramientas de desarrollo de software (parte 2)



El desarrollador **Back-End** es quien trabaja directamente con las solicitudes del cliente. Necesita conocer los **lenguajes de programación**, saber programar las reglas del negocio del sistema y guardar datos.



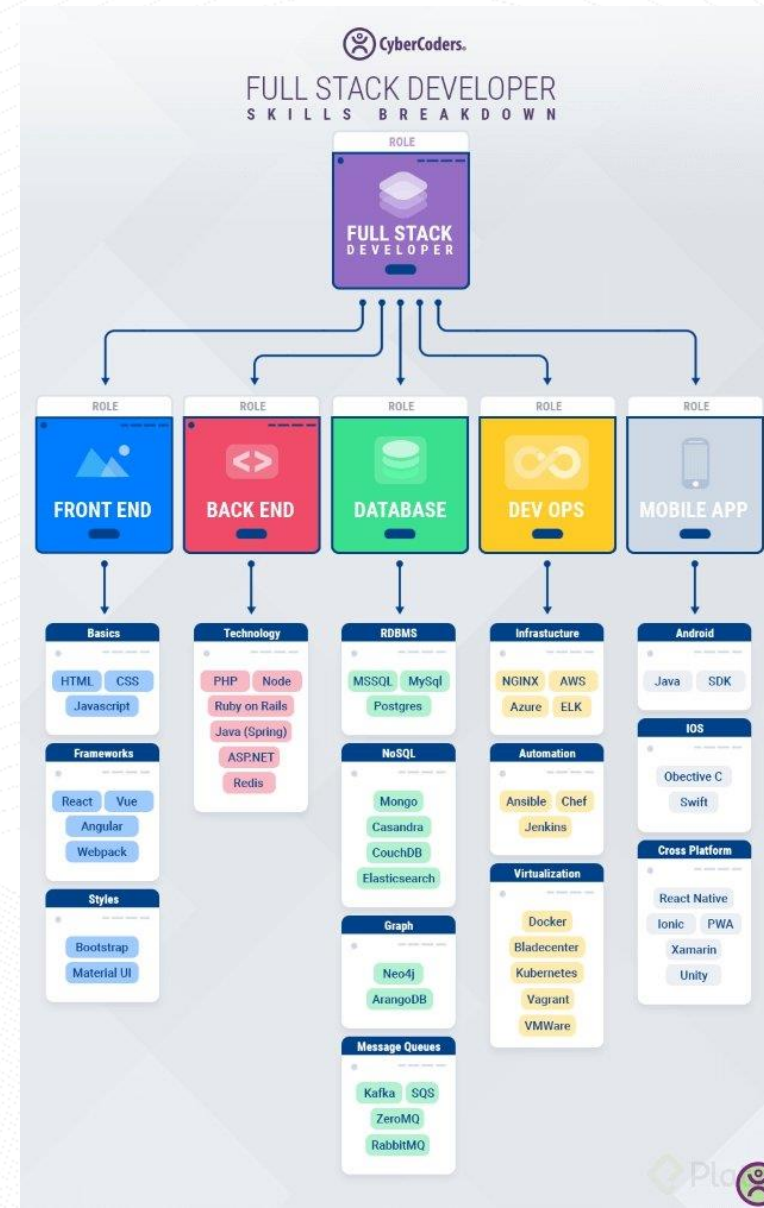
Herramientas de desarrollo de software (parte 3)



Los desarrolladores fullstack se encargan tanto de la parte técnica como de la parte creativa (**frontend y backend**). Un **stack** (en inglés pila o montón) hace referencia al **grupo de tecnologías** que componen un sitio web en todos sus aspectos (desde la base de datos, hacia el manejo lógico y la interfaz visual). Entender un producto de principio a fin.
Ejemplo:



MERN (MongoDB, Express.js, React, Node)
PERN (PostgreSQL, Express.js, React, Node).



Conceptos framework vs librería



Un framework te da toda la estructura para un proyecto completo, desde el inicio hasta el final, integrando funcionalidades sin depender de librerías externas, por ejemplo:

Si necesitas validar formularios, el framework te trae la validación de formularios.
Si necesitas imprimir datos en pantalla, el framework te trae la forma de imprimir esos datos.

FRAMEWORK

Conjunto de herramientas que trabajan en un proyecto completo bajo ciertas reglas.



Tiene funcionalidades integradas para que no necesites librerías externas.



La compatibilidad de sus funcionalidades está asegurada.



El framework define la forma en que debes desarrollar el proyecto.



Laravel



LIBRERÍA

Herramienta con una sola utilidad específica.



Eres libre de usar las librerías que desees en la estructura que quieras.



Debes controlar la compatibilidad de cada librería con las demás.



Puedes usar varias librerías según tus necesidades.



EDgrid

Aprende a programar con las librerías y frameworks más populares en:

ed.team/cursos



Lo genial de las librerías, es que las puedes combinar con otras, por ejemplo: Necesito una librería que me ayude a manejar las fechas (usas la librería de fechas) o necesito una librería que haga drag and drop (usas la librería drag and drop).

Ambos vocablos, «biblioteca» y «librería», son correctos según las definiciones (biblioteca, librería) de la RAE.

Actividades de la sesión



Por grupos de proyecto



Realizar una matriz de comparación de herramientas de desarrollo. Entregar la documentación en .pdf o .xlsx en el espacio asignado.

Back end (Lenguaje de programación):

Herramienta	Logo	Paradigma	¿Tiene licencia? ¿Cuál?	Sistema operativo	Ventaja	Desventaja
PHP						
Java						
JavaScript						
Python						
C#						
Kotlin						

Por grupos de proyecto



Back end (Gestor de base de datos):

Herramienta	Logo	Tipo de base de datos		Sistema operativo	Ventaja	Desventaja
		SQL	NoSQL			
SQLite						
SQLServer						
MySQL						
Oracle						
MongoDB						
PostgreSQL						

Por grupos de proyecto



Front end (vista o interfaz):

Herramienta	Logo	Objetivo	¿Tiene licencia? ¿Cuál?	Sistema operativo	Ventaja	Desventaja
HTML						
CSS						
Bootstrap						
Materialize						
JS						
Angular						
Vue.js						
React						

Reflexión inicial



¿Qué sw
web es
parecido a
su py?

¿Qué
herramientas
tecnológicas
cree que se
usaron?

Realice un pequeño bechmarking (análisis de la competencia) con la siguiente herramienta para averiguarlo: <https://w3techs.com/sites>
Ejemplo: SENA, Panamericana o Facebook.

Sites Info

Find out, which technologies such as CMS, programming language, web server and hosting provider a particular website is using.

Enter url: [Site Info](#)

Examples: [Berkeley.edu](https://berkeley.edu), [Hrw.org](https://hrw.org), [Britannica.com](https://britannica.com)

Arquitectura cliente - servidor



Tienes un ordenador (Smartphone, Tablet, Portátil, etc.) que hace de Cliente y mandas peticiones [1] (a través de una aplicación, navegador web, gestor de correo, etc.) a un Servidor (Servidor físico que tiene Servidores Software), que está al otro lado de "Internet" en la otra punta del planeta,



El cual procesa esta solicitud (un ejemplo de solicitud puede ser querer ver la web de www.Jarroba.com) y devuelve una respuesta [2] (por ejemplo, el contenido que forma la web: texto, imágenes, etc.).

Aplicaciones web (parte 1)

Las aplicaciones web tienen las siguientes características:

- Se **ejecutan en un servidor** (físico o virtual), no en el dispositivo del usuario
- Pueden atender a miles de **usuarios simultáneos**, no a uno sólo
- Es muy habitual que se necesiten varios servidores para una única aplicación web (por escalabilidad y tolerancia a fallos)
- Se ejecutan dentro de un servidor web
- Están **formadas por código y por recursos** (imágenes, documentos html, css, js, ficheros estáticos...)
- Utilizan **servicios adicionales: base de datos**, servidor de correo, servidor de vídeo.
- Requieren de un **proceso de instalación y configuración** en el servidor o servidores.

Empresas de **alojamiento o hosting**: Son empresas que permiten a sus clientes que ejecuten sus aplicaciones web en sus instalaciones.

Aplicaciones web (parte 2)

Las empresas de alojamiento o hosting se encargan de:

- El suministro eléctrico, conexión a Internet, compra y mantenimiento de servidores, control de acceso físico, etc.
- Existe una amplia variedad de servicios que dependen del proveedor y de las necesidades del cliente.

Tipos de alojamiento:

- **Compartido**: Varias aplicaciones web conviven en la misma máquina física. El desarrollador no tiene control total sobre el servidor.
- Dedicado: El cliente dispone de una máquina física que controla completamente.
- Virtual: El cliente dispone de una máquina virtual que controla completamente y se ejecuta en un servidor físico compartido con otras máquinas virtuales.

Alojamiento



Existen muchos proveedores con diferentes servicios (compartido, dedicado y virtual)

Algunos de los más populares:

1 / Más Popular



Hostinger

El **mejor hosting de 2021**. Es rapidísimo, fiable y asequible. Incluye todo lo que necesitamos de forma gratuita y es el único que tiene servidores en LATAM.

Disponibilidad: 99,98 %

Tiempo de carga: 1.0 s

Por cinco dólares al mes, o incluso menos, nos haremos con el mejor paquete de Hostinger: "Empresarial". Este paquete incluye:

- ✓ Centros de datos: **EE.UU., Europa, LATAM y Asia**
- ✓ Copias de seguridad automáticas: **Diarias**
- ✓ Garantía de reembolso: **Treinta días**
- ✓ Almacenamiento en SSD: **200 GB**
- ✓ Certificado SSL: **Gratuito**
- ✓ Dominio: **Gratuito**

3



DreamHost

DreamHost es rápido, fiable e incluye todo lo que necesitamos de forma gratuita. También ofrece un contrato económico por un año.

Disponibilidad: 99,98 %

Tiempo de carga: 1.2 s

Por cinco dólares al mes, o incluso menos, nos haremos con el mejor paquete de DreamHost: "Unlimited". Este paquete incluye:

- ✓ Centros de datos: **EE.UU.**
- ✓ Copias de seguridad automáticas: **Diarias**
- ✓ Garantía de reembolso: **Treinta días**
- ✓ Almacenamiento en SSD: **Ilimitado**
- ✓ Certificado SSL: **Gratuito**
- ✓ Dominio: **Gratuito**

4



HostGator

HostGator no destaca por su rapidez. Pero sus precios son razonables, es fiable e incluye muchas funciones gratuitamente.

Disponibilidad: 99,94 %

Tiempo de carga: 1.4 s

Por cinco dólares al mes, o incluso menos, nos haremos con el mejor paquete de Hostgator: "Turbo". Este paquete incluye:

- ✓ Centros de datos: **EE.UU.**
- ✓ Copias de seguridad automáticas: **Semanales**
- ✓ Garantía de reembolso: **Treinta días**
- ✓ Almacenamiento en SSD: **75 GB**
- ✓ Certificado SSL: **Gratuito**
- ✓ Dominio: **Gratuito**

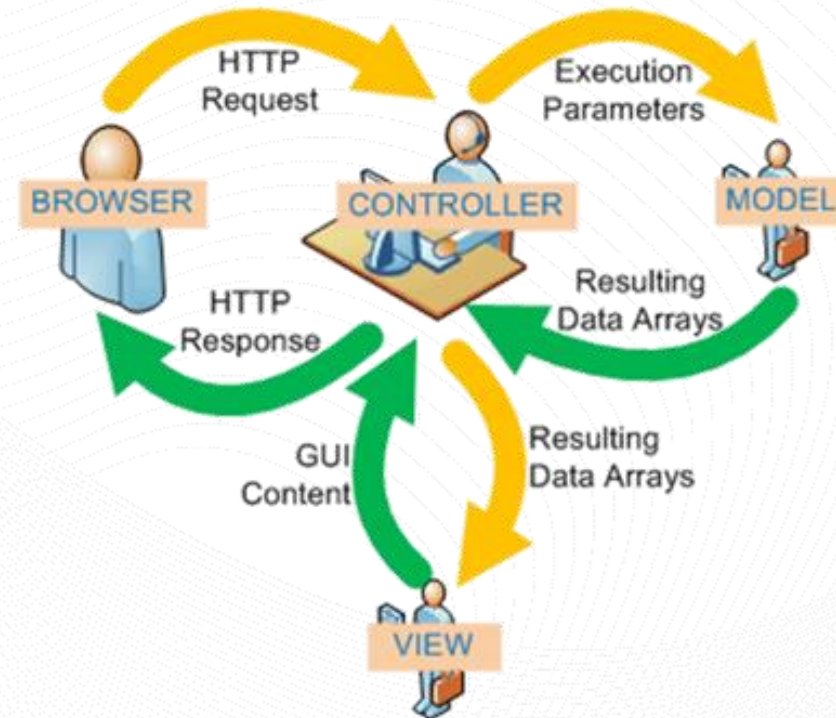
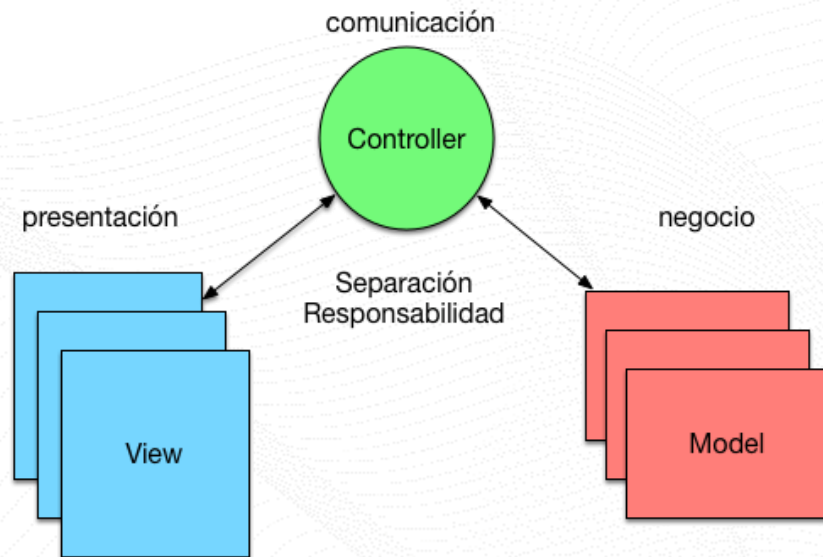
Compare, ¿Cual se adapta a sus necesidades?: [Los mejores hosting en Colombia.](#)

MVC (modelo-vista-controlador)



Se basa en la división de responsabilidades. Es un patrón de arquitectura de software que, utilizando 3 componentes –modelo-vista-controlador- separa la lógica de la aplicación de la lógica de la vista en una aplicación.

(Códigofacilito, 2015)



Reflexión inicial



EDITOR VS IDE



Con ambos puedes escribir código, pero ¿en qué se diferencian?



Software ligero con ayudas para escribir código (resaltado de sintaxis, autocompletado, etc).



Integra un editor con las herramientas que necesita un desarrollador (debugger, compilador, etc).



- Soporta **múltiples lenguajes** y tecnologías.



- **Enfocado en archivos** (no tienen el concepto de proyecto).



- **Puedes agregar plugins** para darle el poder de un IDE pero te toca configurar cada uno a mano.



- Se especializa en **un lenguaje o tecnología** (Java, Python, Go, Android, etc).



- **Enfocado en proyectos completos.** Desde la primera línea hasta la salida a producción.



- Trae **herramientas integradas y configuradas** (ej. Android Studio trae un emulador de Android).

EJEMPLOS DE EDITORES



EJEMPLOS DE IDES



Domina la tecnología con EDteam y #NuncaTeDetengas

👉 ed.team/cursos



Instalación de software

La instalación de los programas de software es el proceso fundamental por el cual los nuevos programas son transferidos a un equipo (computador) con el fin de ser **configurados, y preparados para ser desarrollados**.

Mientras más complejo sea el software, es decir, entre otras características, mientras más archivos contenga, mientras mayor la dispersión de los archivos y mientras mayor sea la interdependencia con otros softwares, mayor es el riesgo de alguna falla durante la instalación.

Por esa razón, sobre todo en casos de software complejo, el desarrollo de un proceso de instalación confiable y seguro es una parte fundamental del equipo desarrollo del software.

La desinstalación de software es el proceso por el que elimina el software del equipo.

Alternativas



Se han producido normas y técnicas cada vez más avanzadas para minimizar y estandarizar el proceso de instalación de software.

Para la instalación de software se pueden aplicar las siguientes técnicas básicas:

- Los **archivos son simplemente copiados en algún lugar del directorio**. Este sistema es fácil e intuitivo. Un riesgo es que versiones más antiguas hayan quedado abandonadas en algún otro lugar sin que nos demos cuenta.
- Se instala primero un **instalador**, el que posteriormente instala el software deseado.
- El sistema operativo o algún software permanente se ocupan de instalar un paquete de software con todos los archivos requeridos. Esto es un Sistema de **gestión de paquetes** (colección de herramientas que sirven para automatizar el proceso, especialmente para GNU/Linux)

Proceso de instalación

- **Verificación de la compatibilidad:** Se debe comprobar si se cumplen los requisitos para la instalación en cuanto a hardware y software. A veces es necesario desinstalar versiones antiguas del mismo software.
- **Verificación de la integridad:** Se verifica que el paquete de software es el original, esto se hace para evitar la instalación de programas maliciosos.
- **Creación de los directorios requeridos:** Para mantener el orden en el directorio cada sistema operativo puede tener un estándar para la instalación de ciertos archivos en ciertos directorios.
- **Creación de los usuarios requeridos:** Para deslindar responsabilidades y tareas se pueden o deben usar diferentes usuarios para diferentes paquetes de software.
- **Concesión de los derechos requeridos:** Para ordenar el sistema y limitar daños en caso necesario, se le conceden a los usuarios solo el mínimo necesario de derechos.

Proceso de instalación



Copiar, desempacar y descomprimir los archivos desde el paquete de software: Para ahorrar ancho de banda y tiempo en la transmisión por internet o espacio de disco duro, los paquetes vienen empacados y comprimidos.

- Archivos principales, sean de fuente o binarios.
- Archivos de datos, por ejemplo datos, imágenes, modelos, documentos XML, etc.
- Documentación
- Archivos de configuración
- Bibliotecas
- Enlaces duros o enlaces simbólico a otros archivos. (Los enlaces duros son copias idénticas del fichero original. Los enlaces simbólicos crean un nuevo inodo (numero entero). Los enlaces duros comparten el número de inodo. En los enlaces simbólicos si borramos el fichero o directorio original, la información se pierde).

Proceso de instalación

- **Compilación y enlace con la bibliotecas requeridas:** En algunos casos no se puede evitar el complicado paso de la compilación y enlace que a su vez tiene severos requerimientos de software al sistema. El enlace con bibliotecas requeridas puede ser un problema si en su instalación no se acataron los estándares establecidos.
- **Configuración:** Por medio de archivos de configuración se le da a conocer al software con que parámetros debe trabajar. Por ejemplo, los nombres de las personas que pueden usar el software, como verificar su clave de ingreso, la ruta donde se encuentran los archivos con datos o la dirección de nuestro proveedor de correo electrónico.
- **Definir las variables de entorno requeridas:** Algunos comportamientos del software solo pueden ser determinados por medio de estas variables. Esto es parte de la configuración, aunque es más dinámica.
- **Registro ante el dueño de la marca:** Para el Software comercial a veces el desarrollador de software exige el registro de la instalación si se desea su servicio.

Término de la instalación



En caso de que las bibliotecas hayan sido cambiadas por la instalación, es necesario arrancar el sistema operativo o el software nuevamente para hacer efectivos los cambios en todos los programas.

Otros aspectos a revisar son:

- Realizar pruebas de funcionamiento general.
- El acceso de otros usuarios al software instalado.

Ejemplo



- Instalación de .NET en Windows: <https://docs.microsoft.com/es-es/dotnet/core/install/windows?tabs=net60>
- Preparación para instalar y configurar software: <https://www.ibm.com/docs/es/bpm/8.5.7?topic=advanced-preparing-install-configure-software>
- Lista técnica de chequeo: <https://www.contraloria.gov.co/documents/342935/383984/Lista-Tecnica-de-Chequeo-V2.pdf/5fb75ff3-b704-51fa-7b5d-d599561f20d1?t=1627573064560&download=true>
- Guía de instalación (antigua)*: https://www.sharp.es/cps/rde/xbcr/documents/documents/Marketing/Operational_manuals/MXC301_ES_ZZ.pdf

Actividad de la sesión



Actividad de la sesión

En grupos de proyecto, realizar una presentación a grandes rasgos del procedimiento de instalación y/o configuración de una de las siguientes plataformas para desarrollo de software:

- Netbeans
- Visual Studio Community
- Visual Studio Code
- IDLE
- SQL Server
- MySQL Workbench
- XAMPP

Al finalizar, deben exponer la información recopilada a sus compañeros de ficha.



GRACIAS

Línea de atención al ciudadano: 01 8000 910270
Línea de atención al empresario: 01 8000 910682



www.sena.edu.co