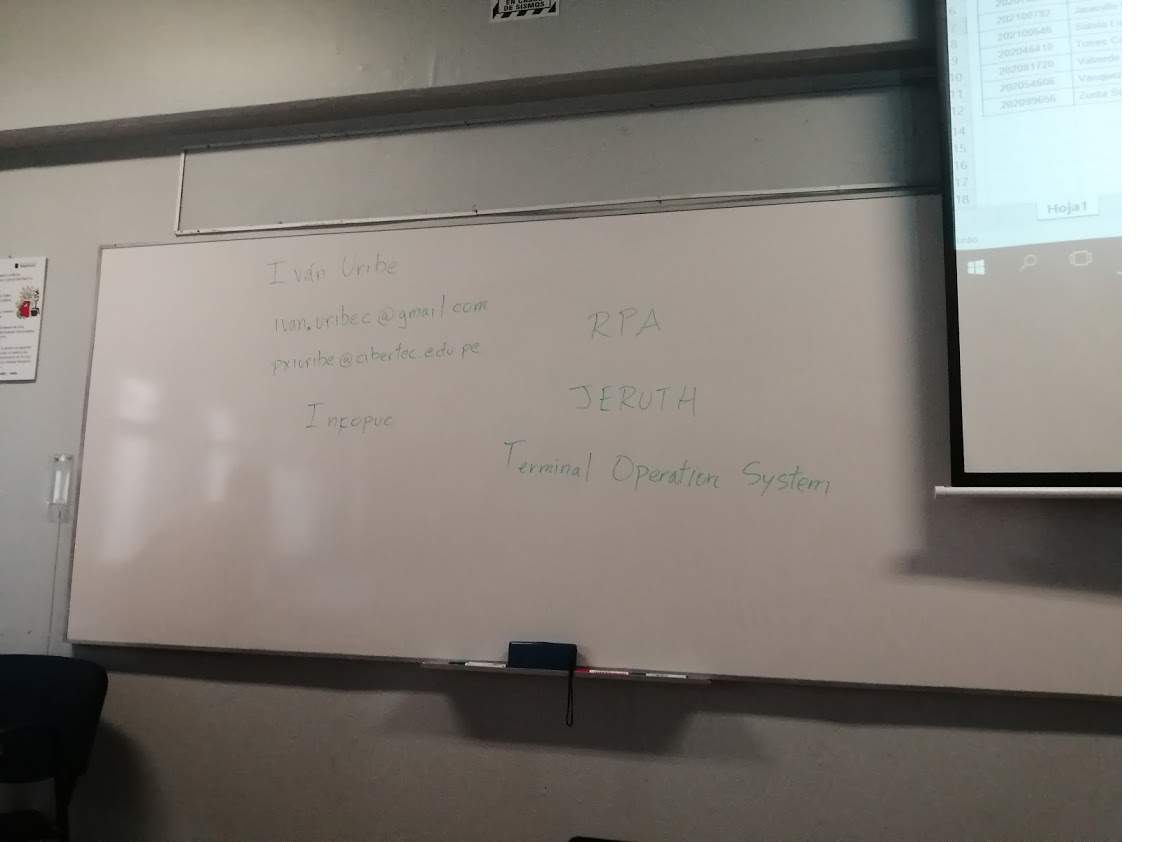
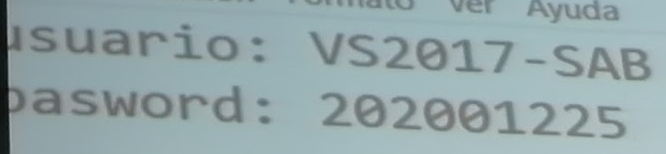
VISUAL STUDIO 2017 WEB DEVELOPER







CAP 01: Desarrollo de Aplicaciones Empresariales

1. Fundamentos de diseño de aplicaciones Web

1. Definición de la arquitectura

Escalabilidad: crecimiento de las transacciones (mayor número de usuarios y tipo de usuarios)

Robustez: la aplicación soporta fallos, datos no consistentes, etc... sin que se pierda la disponibilidad de la aplicación ni los datos.

Ambos están soportados por el Framework que se elija.

Extensibilidad: la aplicación soporta nuevos componentes (extenderse)

Está soportado por el desarrollo. Usar buenos estándares.

Patrones:

De Arquitectura: Define la estructura de los componentes. A nivel de todo el software.

De Diseño: solución a una problemática. Ejemplos (DAO, factory, singleton)

OBS: Ejemplo, el patrón MVC puede ser utilizado como patrón de diseño y también como de arquitectura. Nos damos cuenta cuando observamos la estructura del proyecto. Si MVC está aplicado a todo el proyecto (a nivel de estructura) será patrón de arquitectura y si por el contrario, está aplicado dentro de un componente o clase entonces está aplicado como patrón de diseño.

2. Identificación y distribución del contenido

UX: Accesibilidad-Navegabilidad

3. SEO (Search Engine Optimization)

Buenas prácticas para posicionamiento en buscadores.

4. Personalización:

\* Globalización de la aplicación: tipos de datos como fechas, moneda, etc...

\* Localización: Idioma

5. Configuración:

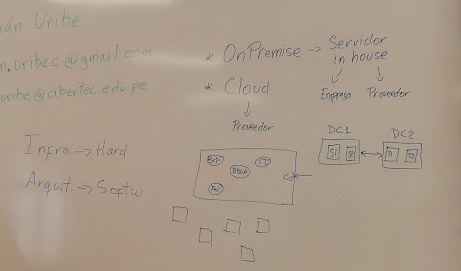
Configuramos la aplicación para diferentes tipos de usuarios.

Introducción a VS 2017:

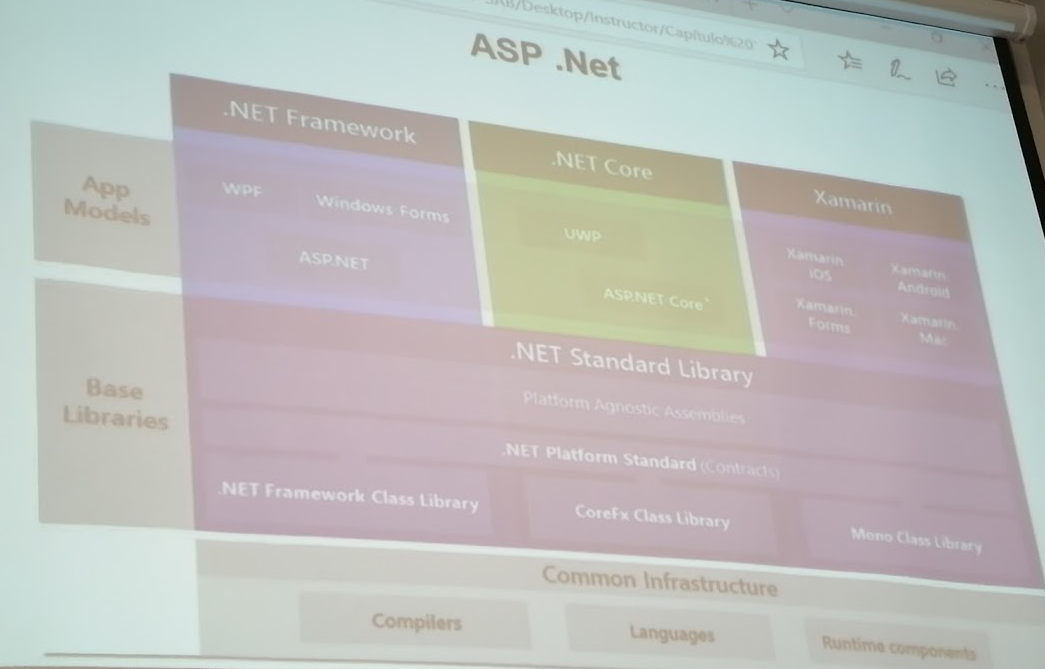
IIS Servidor de aplicaciones, otra opción es Azure.

\* OnPremise -> Servidor in house

\* Cloud -> Servidor en la nube



------------------------------------------------------------------------------------------------------------



Framework

Bibliotecas

Clases

Funcionalidades reutilizables

Estándares

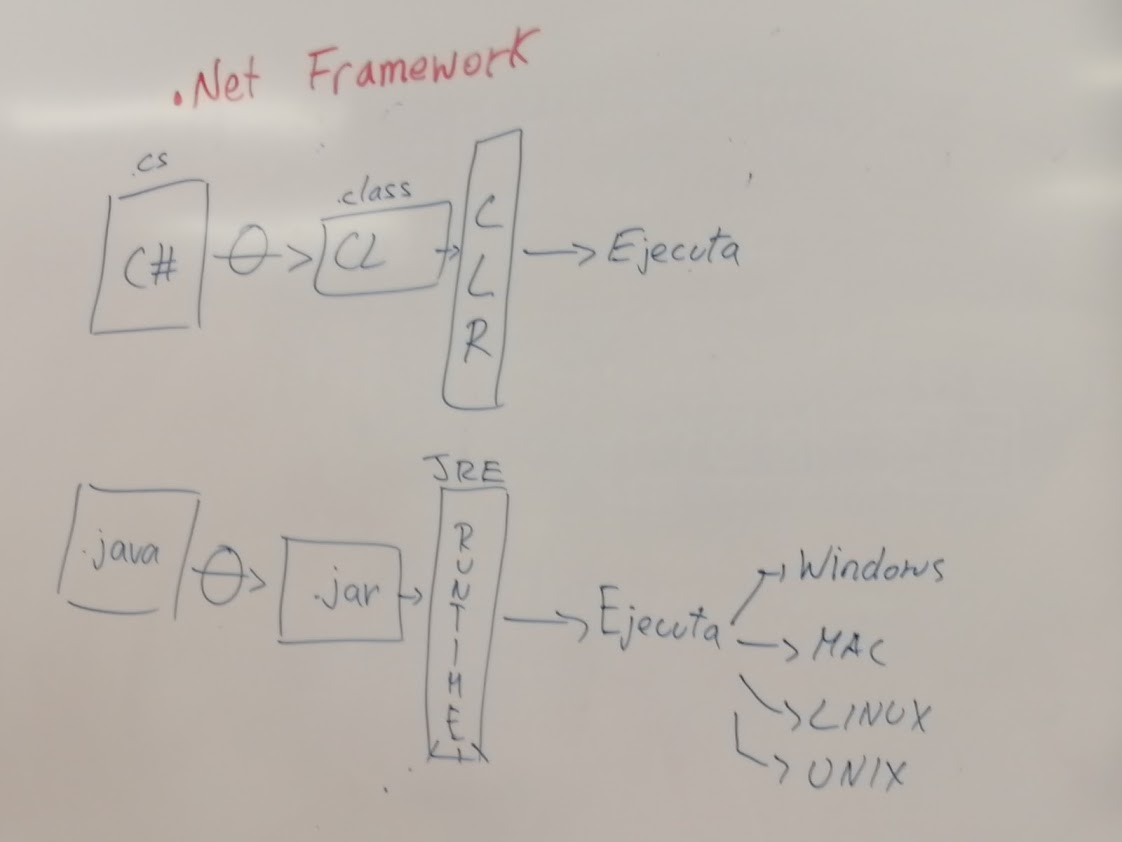
Paquetes

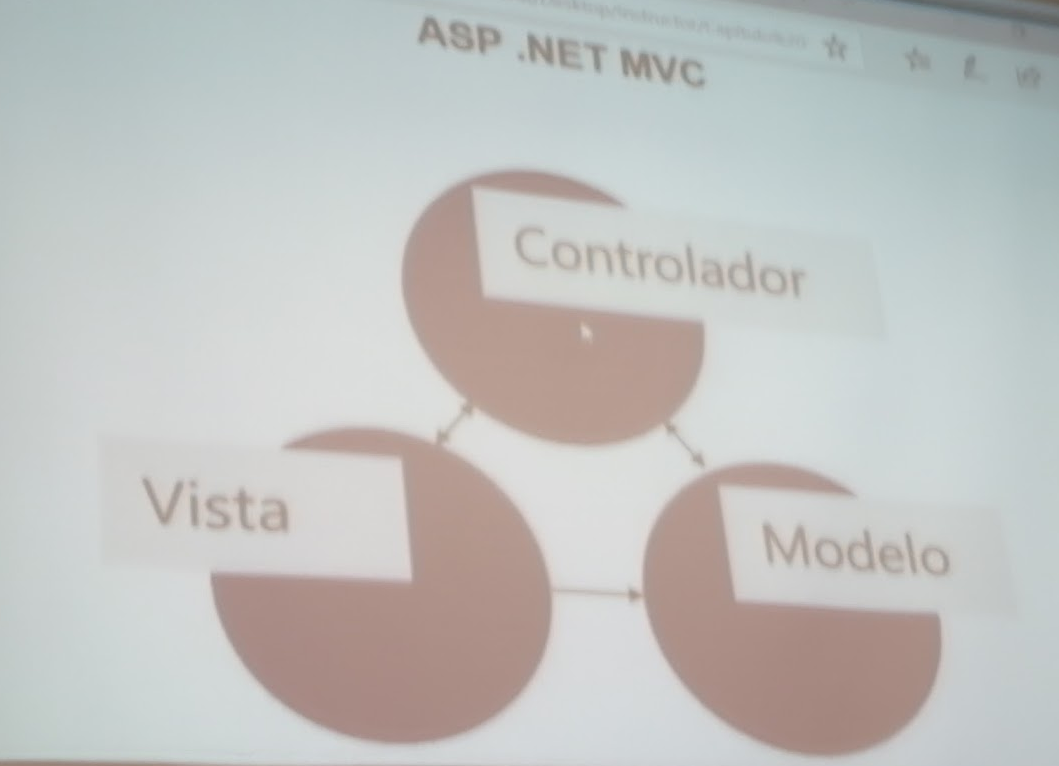
Framework

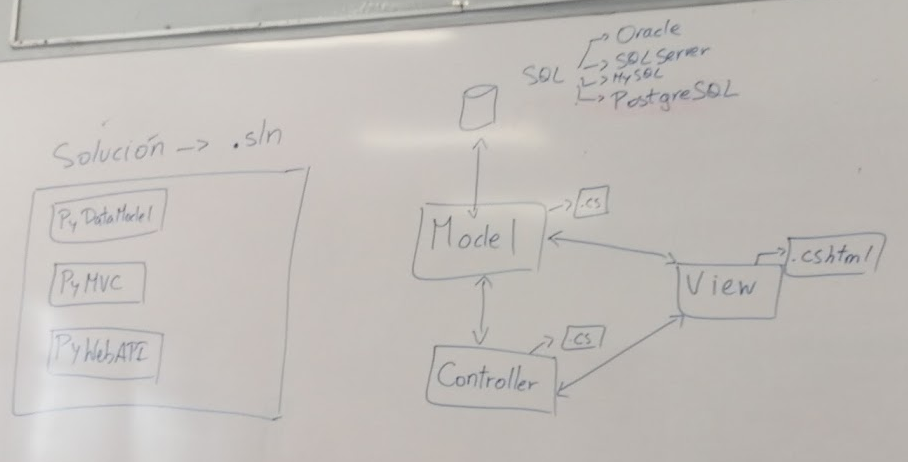
------------------------------------------------------------------------------------------------------------

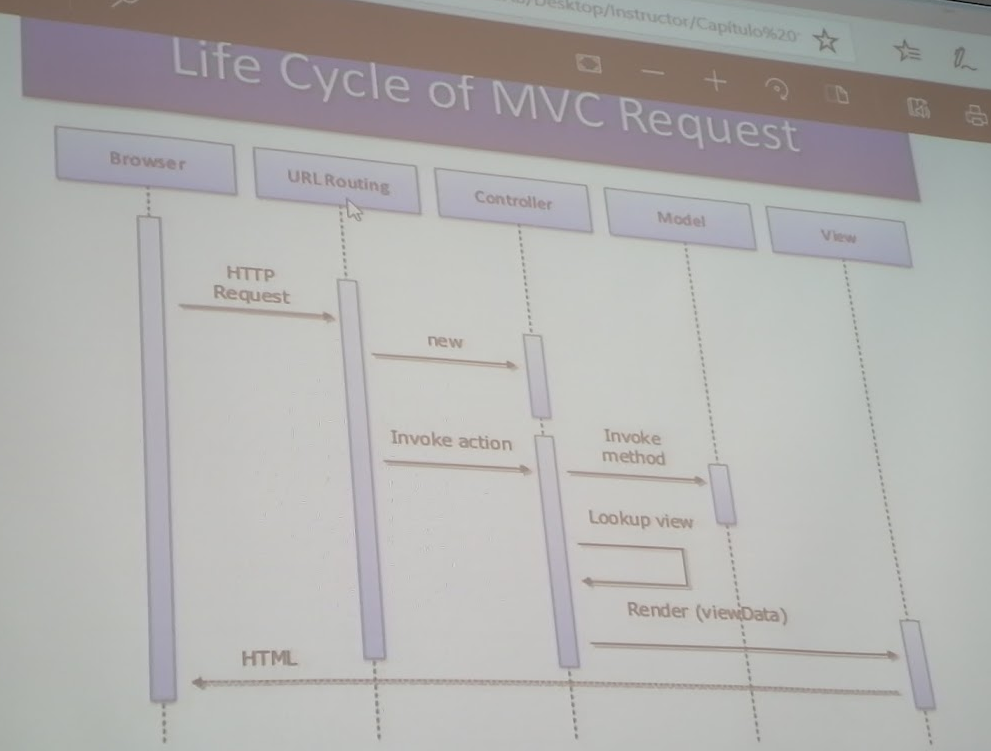
appsettings.json

archivo de configuración. Reemplaza al web.config de .Net Framework









Descargar log4net desde administrador de paquetes NuGet

CAP 02: HTML5 y CSS3

Estructura lógica del proyecto: cómo se ha configurado.

Estructura física del proyecto

SASS

Funciones (mixin) y variables.

Descargar Web compiler desde herramientas/extensiones. Cerrar ventanas VS y ejecutar nuevamente en modo administrador. Se abre el instalador de VSIX y seguir los pasos por defecto.

Generar archivo style.scss fuera de las subcarpetas. Click derecho – Web Compiler – compile file.

En el archivo styles.scss se tiene que tener en cuenta el orden de los @imports. Hay que importar primero los archivos que serán usados por los que se importan después.

15/02/20

CLASE 02

CAPÍTULO 3

Cookies: rastros de las acciones sobre las páginas

Browse caché: elementos de las páginas que se han visitado. Espacio de memoria temporal con dinámica distinta de las cookies.

JAVASCRIPT

Lenguaje interpretado (línea por línea se va ejecutando e interpretando)

Manejo de eventos

Onload: la página se ha cargado completamente.

REST patrón de transferencia

SOAP protocolo, usa XML

SOA patrón de arquitectura

Por buena práctica:

Js de librerías en el head.

Js personalizados abajo para evitar conflictos con las librerías.

Siempre colocar el onload en los scripts.

CAPÍTULO 5

Inyección de Dependencias

Librería SimpleInjector

Instalar paquete nugget: Install-Package SimpleInjector.Integration.Web.Mvc

Crear clase DIConfig y realizar la configuración para la inyección de dependencias del connectionString

El método RegisterMvcController es el que realiza la inyección de dependencias.

Investigar: paradigma programación orientada a aspectos

Log4

Instalar log4 desde nuget

Configurar Glogal.asax

log4net.Config.XmlConfigurator.Configure();

Configurar web.config

<configSections>

<section name="log4net" type="log4net.Config.Log4NetConfigurationSectionHandler, log4net" />

</configSections>

<log4net>

<appender name="RollingFile" type="log4net.Appender.RollingFileAppender">

<file value="C:\Logs\Cibertec.Mvc.log" />

<appendToFile value="true" />

<maximumFileSize value="10000KB" />

<maxSizeRollBackups value="10" />

<layout type="log4net.Layout.PatternLayout">

<conversionPattern value="%date %-5level %logger - %message%newline" />

</layout>

</appender>

<root>

<level value="DEBUG" />

<appender-ref ref="RollingFile" />

<appender-ref ref="aiAppender" />

</root>

</log4net>

Esto es después de la etiqueta configuration

Anotaciones:

Incrustar código DataAnnotations >> Model ( xej [required], [maxLength=50])

Ejecutar en paralelo Action Filters >> Controllers [ActionName = “Agregar”]

Return view retorna la vista renderizandola dentro del layout

Return partialview retorna una vista renderizada directamente (no tiene los estilos del layout)