

.NET

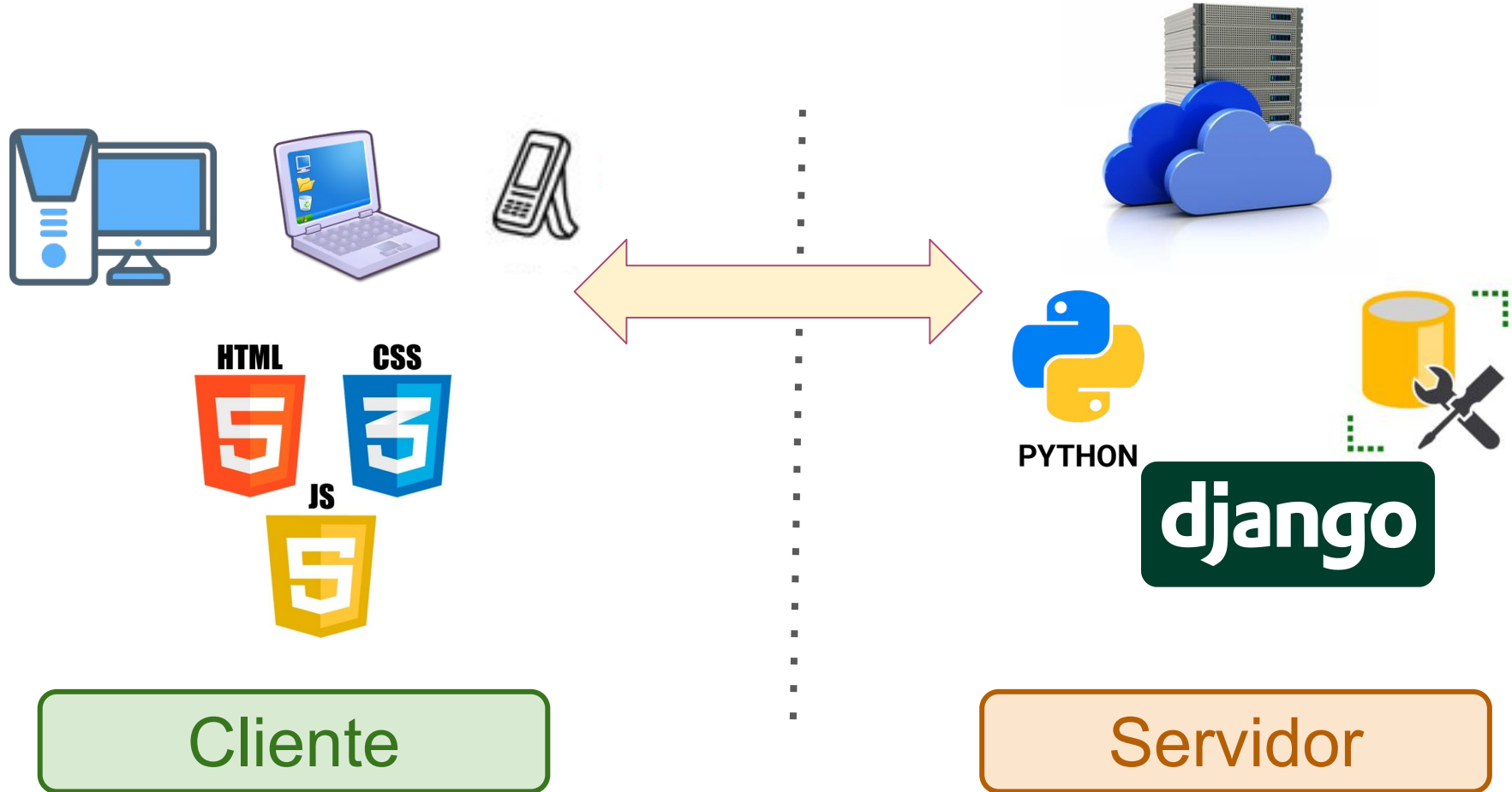
WEB API



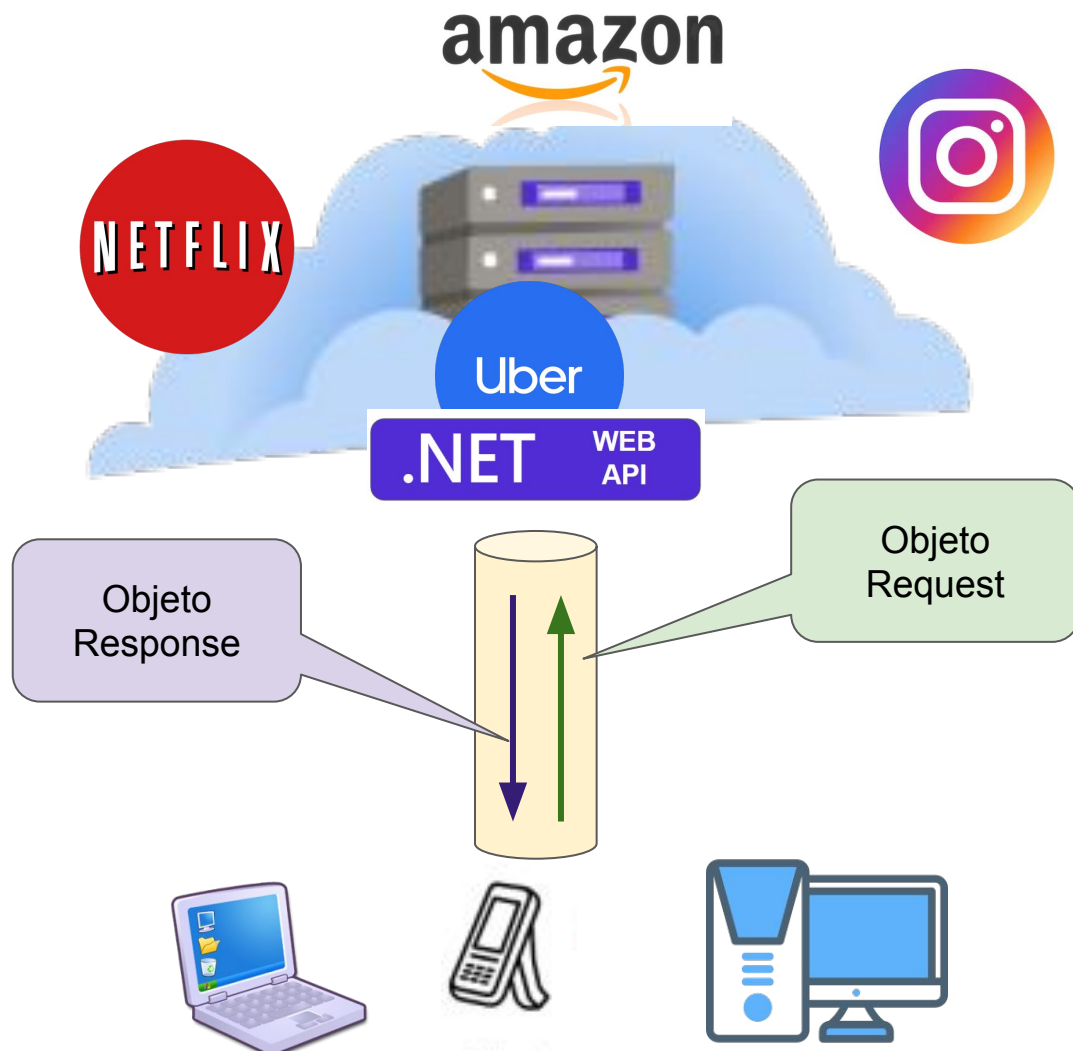
Normalizacion de Base de Datos



Desarrollo Software Fullstack









Modelando el proyecto

Formas Normales

orden_pago	empleado_nombre	salario	fecha_pago	fecha_nacimiento
E001	Juan Perez	1000	11/01/2020	01/01/2000
E002	Juan Perez	1200	15/02/2020	01/01/2001



Modelando el proyecto Formas Normales

Es un proceso para diseñar lógicamente una base de datos



Modelando el proyecto Formas Normales

Ventajas de la normalización:

- Data Redundancy
- Data Integrity
- Database Performance
- Facil de Adaptar



Modelando el proyecto Formas Normales

Que problemas prevenimos con la normalización:

- Anormal Insercion
- Anormal Actualización
- Anormal Eliminacion



Modelando el proyecto

Formas Normales

- 1Forma Normal
- 2Forma Normal
- 3Forma Normal



Modelando el proyecto

Un curso puede ser dictado por múltiples instructores,asimismo un instructor puede dictar diferentes cursos.

Adicionalmente el precio del curso puede variar dependiendo de las promociones que se den, entonces un curso puede tener un precio regular y un precio promocion, este precio puede ser asignado a muchos cursos al mismo tiempo.

Cada curso tiene un puntaje de calidad, dependiendo de este puntaje se puede saber si es un buen curso, recomendado o demasiado complejo.



Nuestro Proyecto

Una app para dar mantenimiento a Cursos, este mantenimiento se basa en :

- Dar de alta nuevos cursos (C)
- Consultar data de cursos (R)
- Actualizar cursos (U)
- Eliminar Cursos (D)



Modelando el proyecto Formas Normales

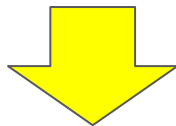
Aplicaremos las 3 formas normales para obtener las entidades y las primary key y foreign key.



Modelando el proyecto

1 Forma Normal

curso_code	curso	calificacion	puntaje	instructor_code	instructor	precio_code	precio
C001	Programacion Basica	buen curso	5	M0001 M0005	Juan Perez Rodrigo Sal	P00101	10
C002	ASP NET Core MVC	Muy complejo	2	M0010	Maria Lopez	P00201 P22140	50 65



curso_code	instructor_code	precio_code	curso	puntaje	comentario	instructor	precio
C001	M0001	P00101	Programacion Basica	5	buen curso	Juan Perez	10
C001	M0005	P00101	Programacion Basica	5	buen curso	Rodrigo Sal	10
C002	M0010	P00201	ASP NET Core MVC	2	Muy complejo	Maria Lopez	50
C002	M0010	P22140	ASP NET Core MVC	2	Muy complejo	Maria Lopez	65



2 Forma Normal

curso_code	instructor_code	precio_code	curso	puntaje	calificacion	instructor	precio
C001	M0001	P00101	Programacion Basica	5	buen curso	Juan Perez	10
C001	M0005	P00101	Programacion Basica	5	buen curso	Rodrigo Sal	10
C002	M0010	P00201	ASP NET Core MVC	2	Muy complejo	Maria Lopez	50
C002	M0010	P22140	ASP NET Core MVC	2	Muy complejo	Maria Lopez	65

curso_code	curso	puntaje	calificacion
C001	Programacion Basica	5	Buen curso
C002	ASP NET Core MVC	2	Muy complejo

instructor_code	curso_code
M0001	C001
M0005	C001
M0010	C002

precio_code	curso_code
P00101	C001
P00201	C002

precio_code	precio
P00101	10
P00201	50
P22140	65

instructor_code	instructor
M0001	Juan Perez
M0005	Rodrigo Sal
M0010	Maria Lopez

3FN

curso_code	curso	puntaje	calificacion
C001	Programacion Basica	5	Buen curso
C002	ASP NET Core MVC	2	Muy complejo

instructor_code	instructor
M0001	Juan Perez
M0005	Rodrigo Sal
M0010	Maria Lopez

instructor_code	curso_code
M0001	C001
M0005	C001
M0010	C002

precio_code	curso_code
P00101	C001
P00201	C002

precio_code	precio
P00101	10
P00201	50
P22140	65

curso_code	curso
C001	Programacion Basica
C002	ASP NET Core MVC

puntaje	calificacion
1	Bajo nivel
2	Muy complejo
3	Medio curso
4	Curso recomendado
5	Buen Curso

precio_code	curso_code
P00101	C001
P00201	C002

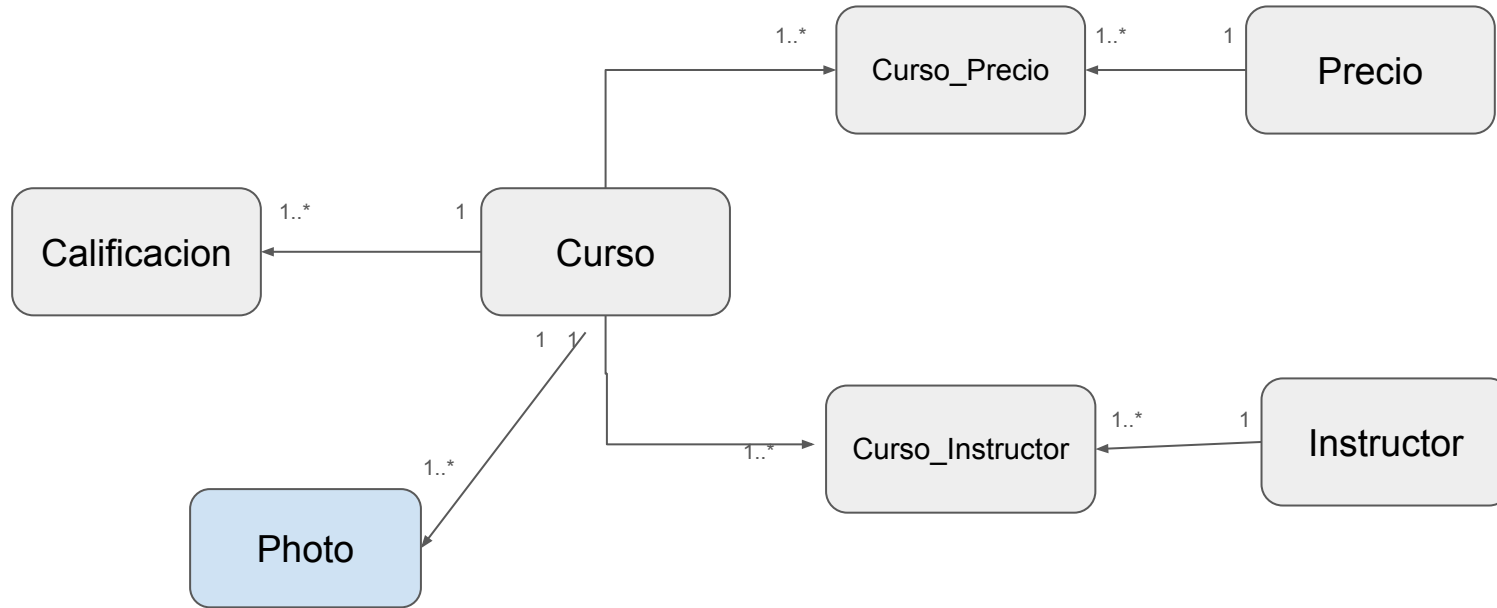
instructor_code	instructor
M0001	Juan Perez
M0005	Rodrigo Sal
M0010	Maria Lopez

instructor_code	curso_code
M0001	C001
M0005	C001
M0010	C002

precio_code	precio
P00101	10
P00201	50
P22140	65



Modelando el proyecto

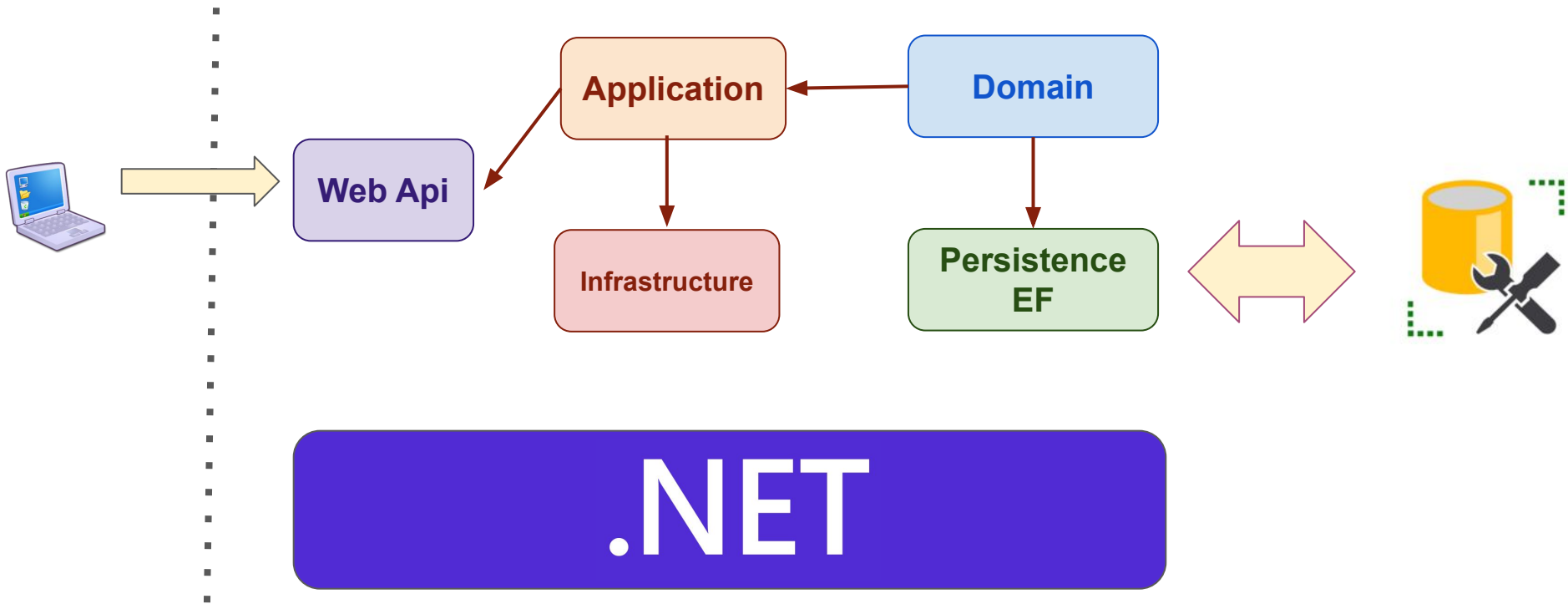




WEB APIs con .NET

Cliente

Servidor





WEB API vs REST

Que es REST?

- En el 2000 se publicó una tesis en US:
ARCHITECTURAL STYLES IN THE DESIGN OF NETWORK-BASED SOFTWARE ARCHITECTURES
- Aqui se definio el termino REST

“REST se basa en operar las representaciones de recursos de un computador utilizando hypermedia”

REST y Web Apis?

- Muchos de los conceptos de Web Apis han sido tomados de REST
- El precursor de Web Apis es REST

Que es REST?

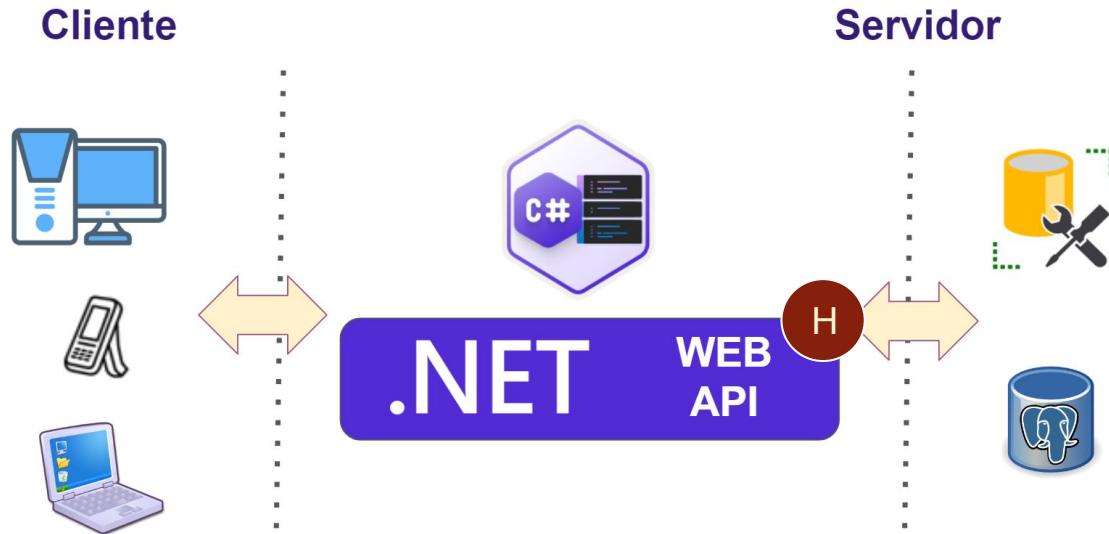
Que significa:

REPRESENTATIONAL **S**TATE **T**RANSFER

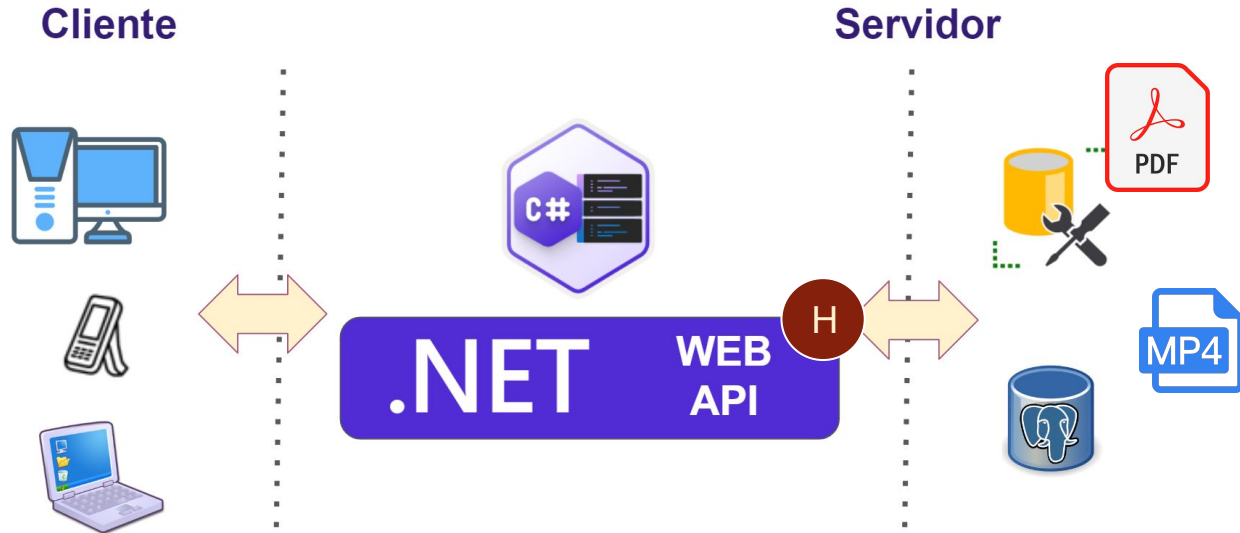
La definición del creador de REST:

“UN ESTILO DE ARQUITECTURA PARA CONSTRUIR SISTEMAS DISTRIBUIDOS BASADOS EN HIPERMEDIA Y DISEÑADO SOBRE RECURSOS”

Que es Hypermedia?

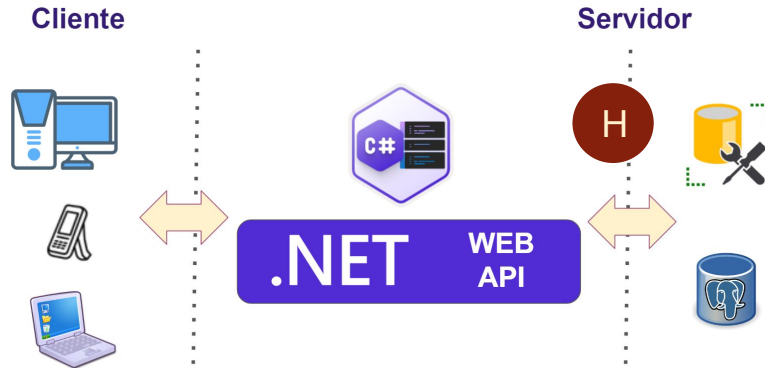


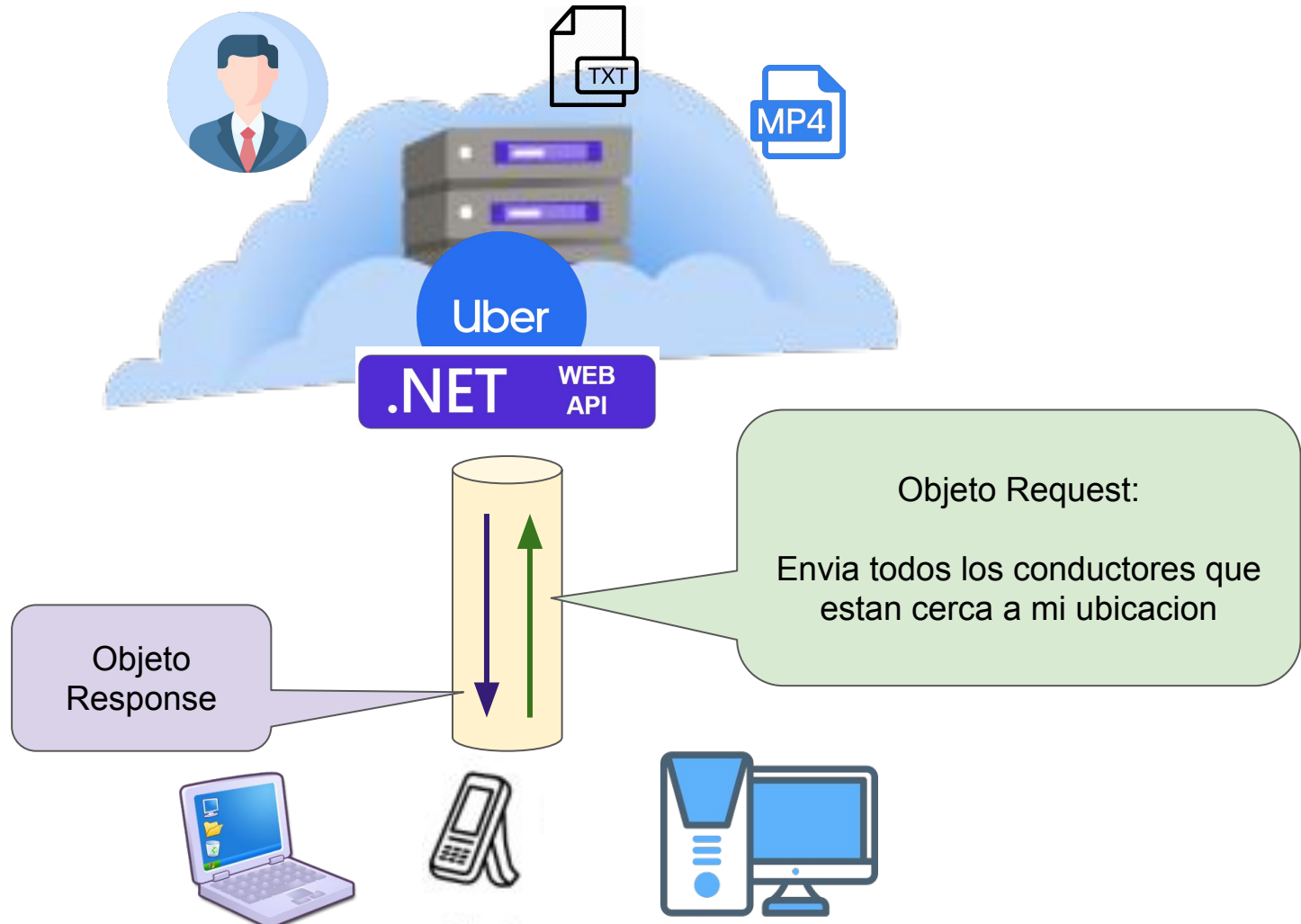
Que Recursos?

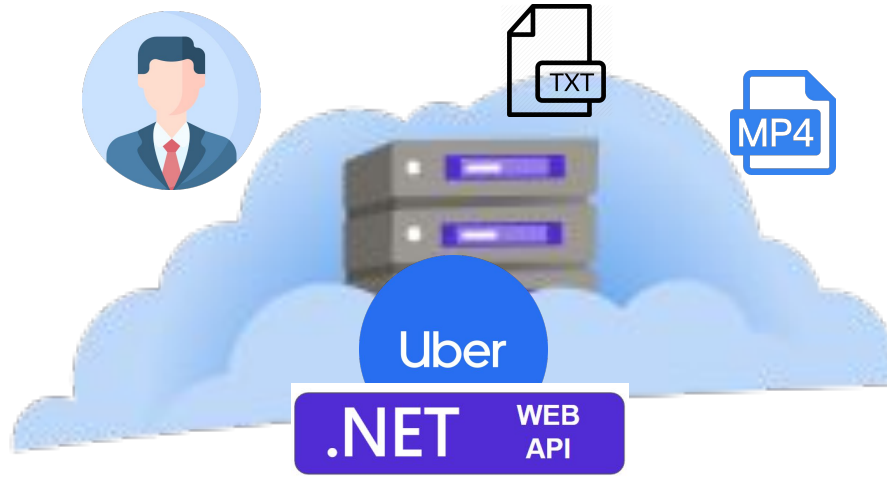


Que es un URI?

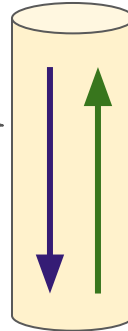
<http://www.vaxidrez.com/archivos/file.pdf>







La transmisión se da en
formato bytes



Diseñando URIS para los Recursos

Usa nombres en vez de verbos:

Ejemplo:

Evita: /agregarUsuario

Usa: /usuarios

Diseñando URIS para los Recursos

Usa Plural en vez de Singular

Ejemplo:

Evita: /curso

Usa: /cursos

Diseñando URIS para los Recursos

Agrega un identificador

Ejemplo:

Usa: /cursos/1

Evita Exponer Detalles Internos

Esquema de la base de datos

Persistencia de los modelos

Business domain entities

HTTP Verbos / Metodos

GET

POST

PUT

DELETE

PATCH

HTTP Verbos Adicionales

HEAD

OPTIONS

Http Status Code Classes

100-199

Informational

200-299

Successful

300-399

Redirection

400-499

Client Error

500-599

Server Error

Escenarios de Transaccion

Obtener un
Recurso

Crear un
Recurso

Modificar
un Recurso

Eliminar un
Recurso

Seguridad/
Permisos

Excepcion
unhandled

Obtener un Recurso (GET)

200 OK

301 Moved Permanently

302 Found

304 Not Modified

404 Not Found

429 Too Many Requests (RFC 6586)

Crear un Recurso (POST)

200 OK

201 Created

202 Accepted

204 No Content

303 See Other

400 Bad Request

429 Too Many Requests

Modificar un Recurso (PUT, PATCH)

200 OK

201 Created (see RFC 7231 6.3.2)

202 Accepted

204 No Content

400 Bad Request

404 Not Found

409 Conflict

429 Too Many Requests

Eliminar un Recurso (DELETE)

200 OK

201 Created (see RFC 7231 6.3.2)

202 Accepted

204 No Content

400 Bad Request

404 Not Found

409 Conflict

429 Too Many Requests

Seguridad / Permisos

401 Unauthorized

403 Forbidden

404 Not Found

Errores Unhandled Exceptions

500 Server Error

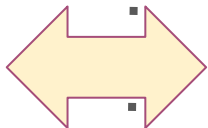
Tipos de Response en Web Api

Tipo Especifico -> string - List<int> - Curso

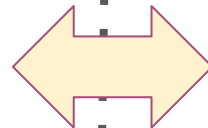
IActionResult -> Ok() - NotFound() - BadRequest()

ActionResult<T> -> Ok() - NotFound() - BadRequest()

Cliente

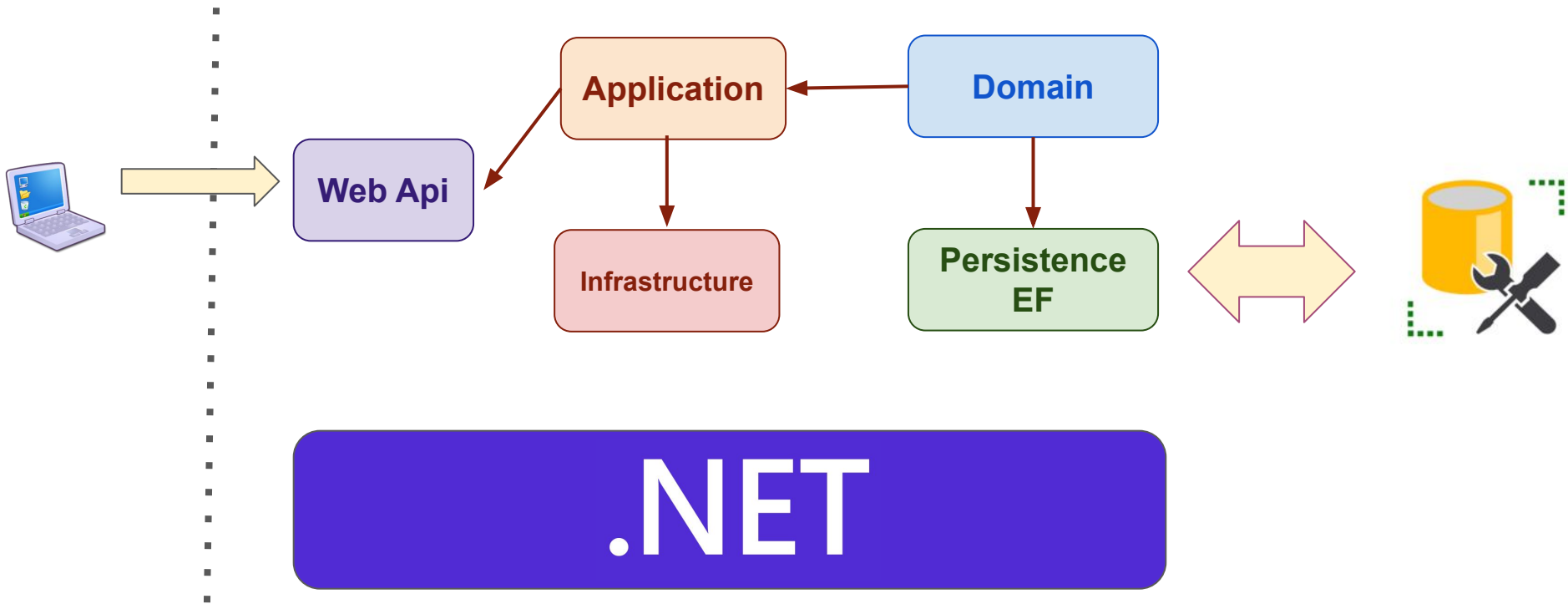


Servidor



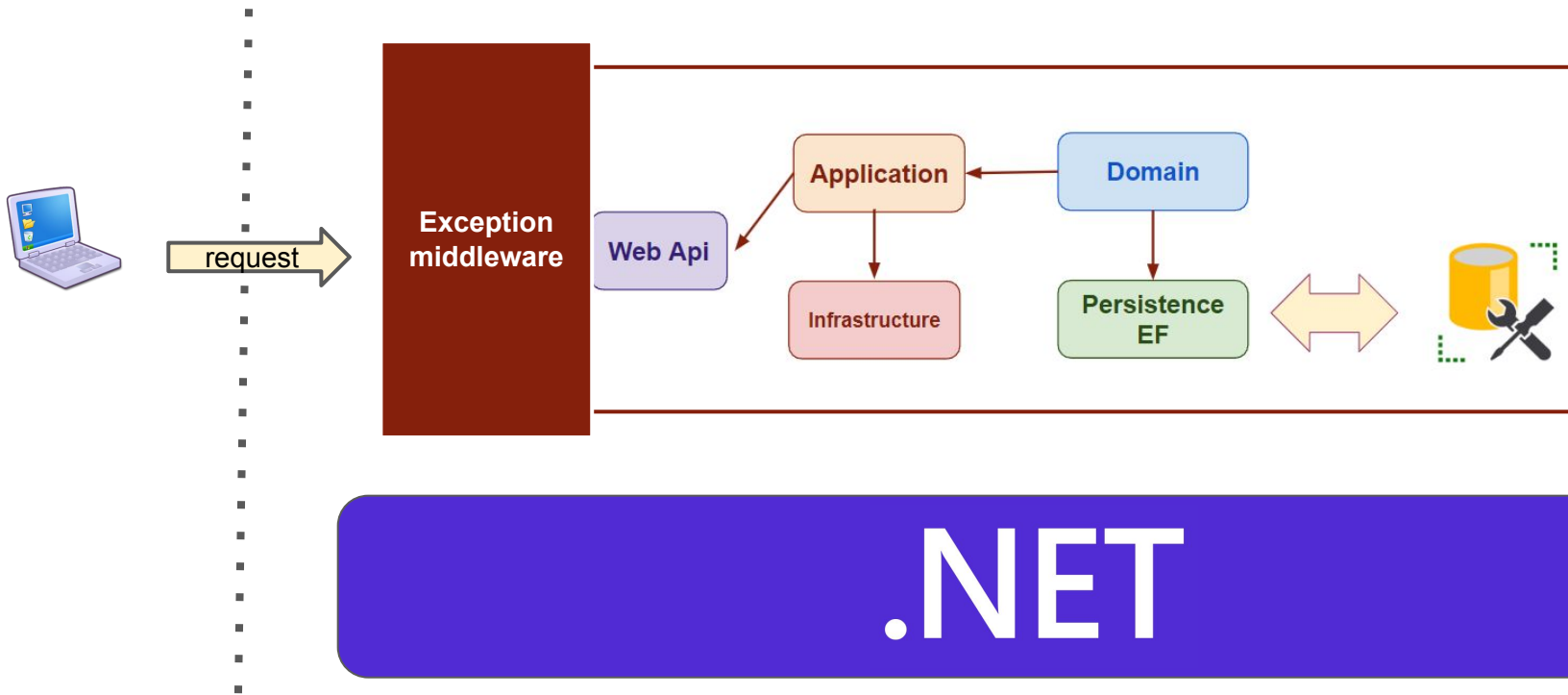
Cliente

Servidor



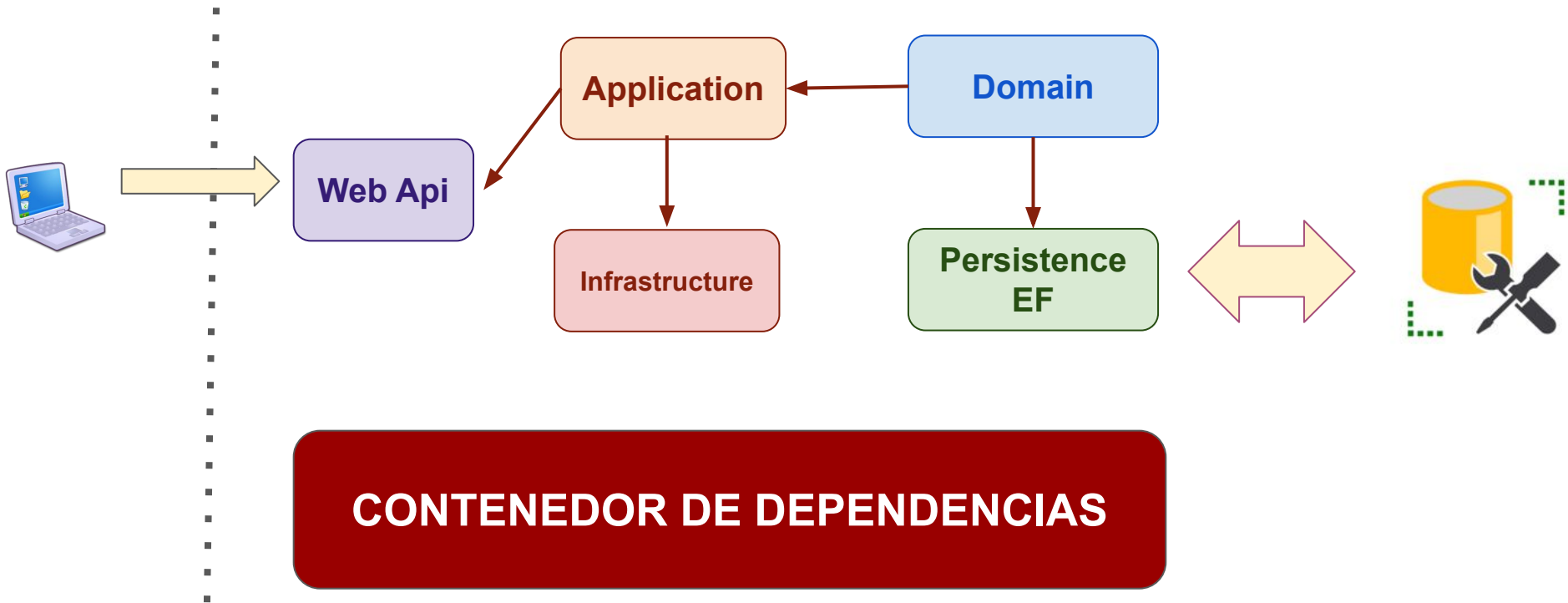
Cliente

Servidor



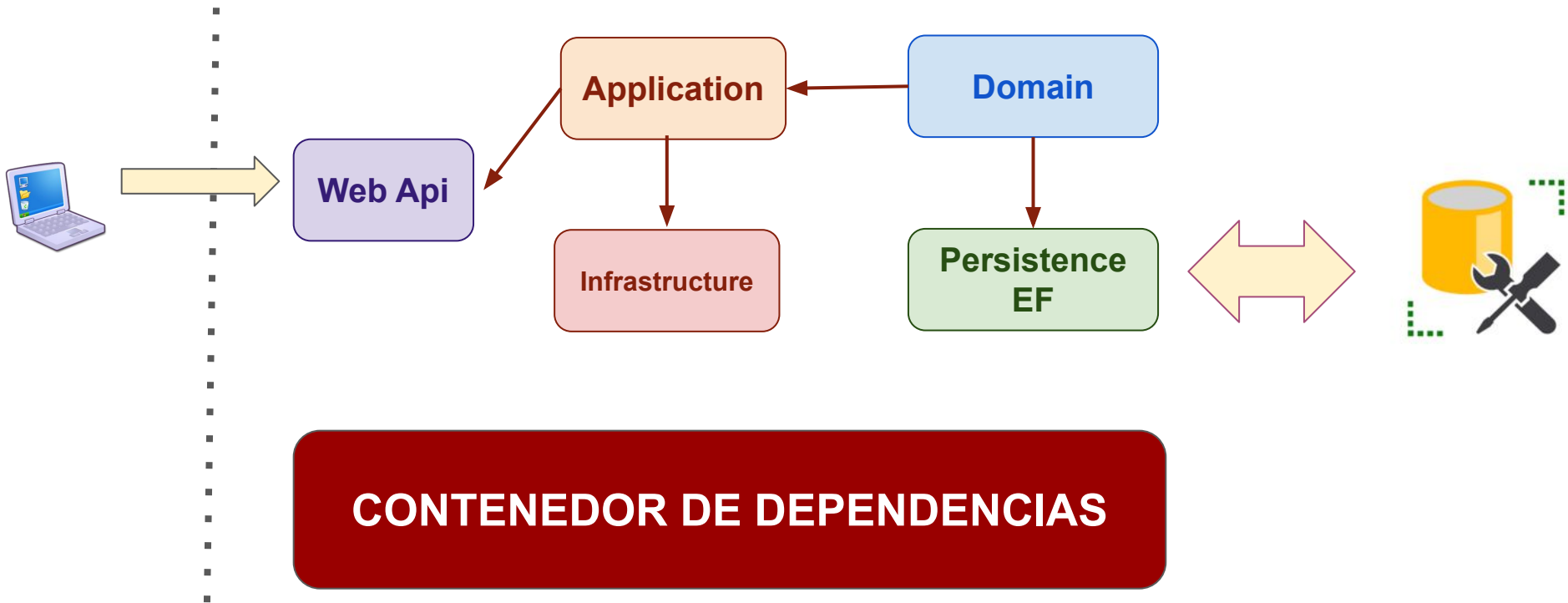
Cliente

Servidor



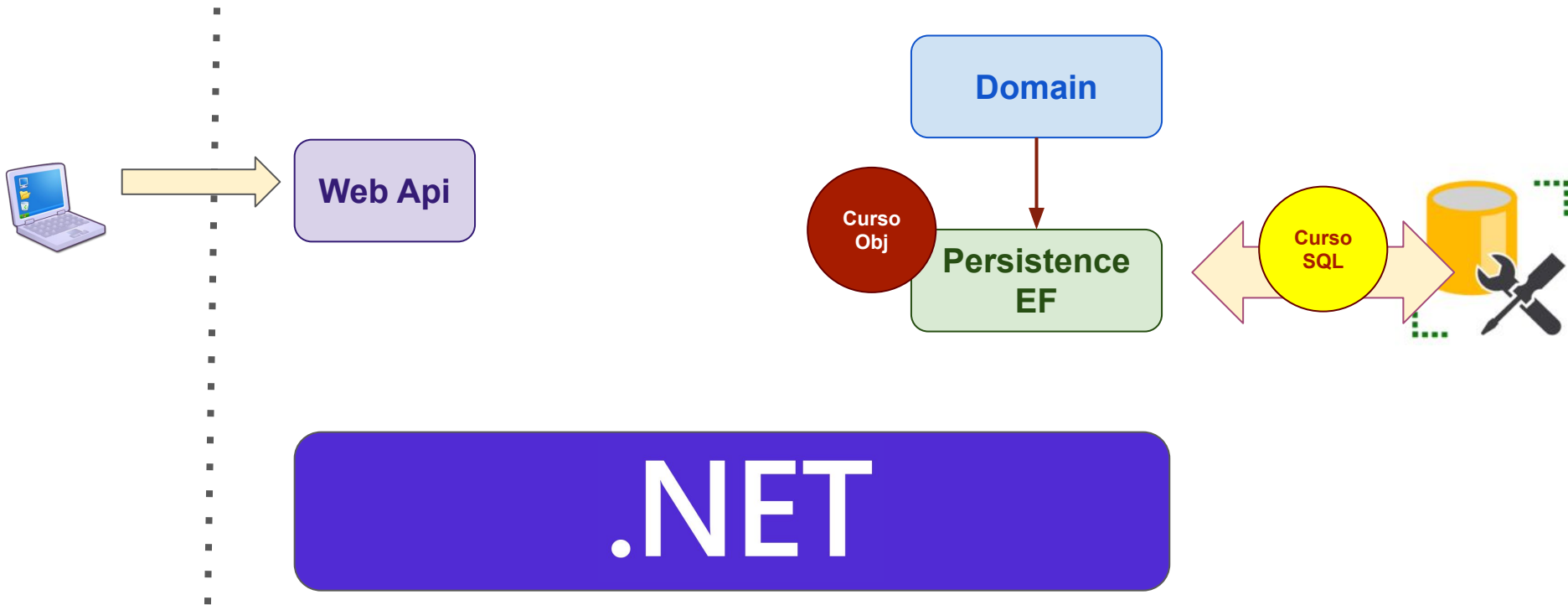
Cliente

Seguridad



Cliente

Servidor



Identity Core

