

AYUDANTÍA PROYECTO

DIEGO IRURETAGOYENA
DIIRU@UC.CL
IIC2413 – BASES DE DATOS
28 MARZO, 2018

OBJETIVOS

ENFOQUE EN LAS ENTREGAS 2, 3 Y 4
EL OBJETIVO ES APRENDER:

- **ESTRUCTURA WEB:** PARA ENTENDER EL FUNCIONAMIENTO.
- **HTML:** PARA CREAR LA VISUALIZACIÓN DE LA PÁGINA.
- **PHP:** PARA HACER LA CONEXIÓN CON LA BASE DE DATOS Y HACER CONSULTAS.

TODO EL CONTENIDO VISTO DENTRO DE ESTA AYUDANTÍA SOLO SERÁ EVALUADO DENTRO DE LAS ENTREGAS DE PROYECTO RESTANTES. NO SERÁ REQUISITO PARA LAS INTERROGACIONES NI PARA EL EXAMEN

ESTE CURSO NO ES DE WEB, POR LO QUE NO SERÁ EVALUADO DENTRO DE LAS ENTREGAS. ESTA PARTE ES SOLO ENTENDER LA DINÁMICA DE LA PÁGINA. OTROS CURSOS WEB:

IIC2143: INGENIERÍA DE SOFTWARE
IIC2513: TECNOLOGÍAS Y APLICACIONES WEB

1. VISUALIZACIÓN

- ▶ HTML
- ▶ CSS
- ▶ JS

2. LÓGICA DE LA APLICACIÓN

- ▶ PHP
- ▶ BASE DE DATOS

COMPONENTES A UTILIZAR

DE LOS COMPONENTES DESCRITOS, SOLO USAREMOS **HTML, PHP Y BASE DE DATOS**.
CSS SE USA PARA TENER UNA VISUALIZACIÓN MÁS ATRACTIVA MIENTRAS QUE
JAVASCRIPT PARA LA DINÁMICA DE LA PÁGINA (EJ. ANIMACIONES).

ESTAS COMPONENTES SON IDEALES, **PERO NO REQUISITOS**.

HTTP

ADEMÁS DE LOS COMPONENTES PREVIOS, ES NECESARIO QUE ENTENDAMOS LO QUE ES EL

HYPertext TTransfer PROTOCOL – PROTOCOLO DE TRANSFERENCIA DE HIPERTEXTO

PROTOCOLO UTILIZADO PARA TRANSFERIR INFORMACIÓN EN LA WORLD WIDE WEB.
EL BROWSER (NAVEGADOR) DEBE COMUNICARSE CON EL SERVIDOR WEB A TRAVÉS DE ESTE PROTOCOLO. PARA ELLO,
EN LA CONEXIÓN SE ESPECIFICA EL HOSTNAME Y EL PORT NUMBER.

LA SOLICITUD QUE EL NAVEGADOR LE ENVÍA INCLUYE UN **MÉTODO** Y UN **RECURSO**.

ESTRUCTURA DEL HTTP REQUEST

MÉTODO + URI* DEL RECURSO QUE QUEREMOS UN EJEMPLO DE ESTO SERÍA:

GET HTTP://LOCALHOST:3000/SERIES/3

EN ESTE EJEMPLO, SE LE SOLICITA AL SERVIDOR QUE ENTREGUE LA SERIE (EN SERIES) DE IDENTIFICADOR 3. ANALIZANDO LA SOLICITUD, TENEMOS QUE:

- **GET** ES EL MÉTODO DEL REQUEST.
- **HTTP** ES EL ESQUEMA* DE LA URI.
- **LOCALHOST** ES EL HOSTNAME.
- **3000** ES EL PORTNUMBER.
- **SERIES/3** ES LA RUTA QUE EL SERVIDOR DEBE SEGUIR. *PARA MÁS INFORMACIÓN, VER **ESTO**.

MÉTODOS MÁS UTILIZADOS DEL HTTP REQUEST

- ▶ **GET:** LE SOLICITA AL SERVIDOR UN RECURSO (GENERAL O ESPECÍFICO). EJEMPLO: GET HTTP://LOCALHOST:3000/SERIES
- ▶ **POST*:** LE SOLICITA AL SERVIDOR ALMACENAR LA INFORMACIÓN ENVIADA COMO UNA NUEVA ENTIDAD DEL RECURSO EN LA URI.
EJEMPLO: POST HTTP://LOCALHOST:3000/SERIES
- ▶ **PUT*:** LE SOLICITA AL SERVIDOR ALMACENAR UNA ENTIDAD EN LA UBICACIÓN ESPECÍFICA ENVIADA.
EJEMPLO: PUT HTTP://LOCALHOST:3000/SERIES/3
- ▶ **PATCH:** LE SOLICITA AL SERVIDOR EDITAR LA ENTIDAD ALMACENADA EN LA UBICACIÓN ESPECÍFICA ENVIADA.
EJEMPLO: PATCH HTTP://LOCALHOST:3000/SERIES/3
- ▶ **DELETE:** LE SOLICITA AL SERVIDOR ELIMINAR LA ENTIDAD ALMACENADA EN LA UBICACIÓN ESPECÍFICA ENVIADA.

EJEMPLO: DELETE HTTP://LOCALHOST:3000/SERIES/3

*DISTINCIÓN ENTRE POST Y PUT **AQUÍ.**

ESTRUCTURA DE LA CONEXIÓN



ESTRUCTURA DE LA CONEXIÓN

1. EL BROWSER SE CONECTA CON EL SERVIDOR, Y LE ENVÍA LA SOLICITUD HTTP.
2. EL SERVIDOR, A TRAVÉS DE LA SOLICITUD, OBTIENE LA RUTA Y EJECUTA LA ACCIÓN ASOCIADA A ELLA.
3. EJECUTADA LA ACCIÓN, EL SERVIDOR LE RETORNA AL BROWSER LA VISUALIZACIÓN EN FORMA DE HTML (A TRAVÉS DE UNA RESPUESTA HTTP).
4. FINALMENTE, EL BROWSER RECIBE LA RESPUESTA HTTP, LA PROCESA Y GENERA LA REPRESENTACIÓN VISUAL QUE SERÁ VISTA POR EL USUARIO.

ESTA ES UNA VERSIÓN RESUMIDA DEL PROCESO. NUEVAMENTE, ESTE CURSO NO SE ENFOCA EN ESTE TEMA, ESTO ES CONTENIDO DE REFERENCIA PARA ENTENDER LA LÓGICA QUE HAY DETRÁS.

HTML

HYPERT**EX**T **MA**RKUP **LA**NGUAGE – LENGUAJE DE MARCAS DE HIPERTEXTO

- ▶ ES EL LENGUAJE DE MARCADO ESTÁNDAR UTILIZADO PARA CREAR PÁGINAS WEB.
- ▶ SERÁ LA REPRESENTACIÓN VISUAL DE LA PÁGINA.
- ▶ SE PUEDE EDITAR DESDE CUALQUIER EDITOR DE TEXTO, PERO SE RECOMIENDA UTILIZAR **SUBLIME TEXT**, O BIEN **ATOM** (ENLACES DE DESCARGA EN EL NOMBRE DE CADA UNO).

<MARCA>CONTENIDO**</MARCA>**

MARCA ESTÁNDAR. **MARCA** VIENE A SER EL TIPO DE MARCA, Y **CONTENIDO** LO QUE ESTÁ DENTRO DE ELLA. TODA MARCA VIENE DE A PARES (**<MARCA>** QUE INDICA EL INICIO, Y **</MARCA>** QUE INDICA EL FINAL), AUNQUE HAY ALGUNOS ESPECÍFICOS QUE PUEDEN TENER UN ÚNICO ELEMENTO (POR

EJEMPLO,

**
** QUE ES UN SALTO DE LÍNEA).

TIPOS DE MARCA

A CONTINUACIÓN, SE VERÁ UN LISTADO Y EJEMPLOS APLICADOS PARA VER LOS TIPOS DE MARCA QUE UTILIZAREMOS EN NUESTRA PÁGINA.

TIPOS DE MARCA - BÁSICAS - EJEMPLOS - HTML MÍNIMO

- ▶ **HTML:** MARCA QUE DESCRIBE EL DOCUMENTO HTML (TODAS LAS MARCAS VAN DENTRO DE ÉL).
- ▶ **HEAD:** MARCA QUE CONTIENE EL ENCABEZADO (INFORMACIÓN SOBRE EL DOCUMENTO).
- ▶ **TITLE:** MARCA QUE CONTIENE EL TÍTULO DE TODA LA PÁGINA.
- ▶ **BODY:** MARCA QUE CONTIENE EL CUERPO DEL DOCUMENTO. DENTRO DE ESTE INSERTAMOS TODOS LOS ELEMENTOS QUE QUEREMOS AÑADIR A NUESTRA VISUALIZACIÓN.
- ▶ **H1:** MARCA QUE CONTIENE UN TÍTULO, INDICANDO UNA SECCIÓN DEL DOCUMENTO. AUMENTANDO EL NÚMERO, SE CONVIERTE EN UNA SUBSECCIÓN (POR EJEMPLO, **H2** SERÁ LA SUBSECCIÓN DE **H1**, **H3** SERÁ LA SUBSECCIÓN DE **H2**, Y ASÍ SUCESIVAMENTE).
- ▶ **P:** MARCA QUE CONTIENE UN PÁRRAFO DE TEXTO.
- ▶ **<!--..-->:** MARCA PARA DEJAR COMENTARIOS (DONDE VAN LOS "..").

TIPOS DE MARCA - BÁSICAS - EJEMPLOS - HTML MÍNIMO

```
<!DOCTYPE HTML>  
<HTML>  
<HEAD>  
<TITLE>EJEMPLO 1</TITLE>  
</HEAD>  
<BODY>  
</BODY>  
</HTML>
```

CÓDIGO MÍNIMO QUE DEBE ESTAR CONTENIDO EN EL DOCUMENTO PARA QUE EXISTA UNA VISUALIZACIÓN.

EJEMPLO

```
<!-- SOY UN COMENTARIO -->
<!DOCTYPE HTML>
<HTML>
<HEAD>
<!-- CON ESTO, PODEMOS VER LOS TILDES -->
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="TEXT/HTML; CHARSET=UTF-8" />
<TITLE>EJEMPLO 3</TITLE>
</HEAD>
<BODY>
<H1> SOY LA SECCIÓN 1 </H1>
<P> SOY UN PÁRRAFO DE LA SECCIÓN 1 </P>
<H2> SOY LA SECCIÓN 2 </H2>
<P> SOY UN PÁRRAFO DE LA SECCIÓN 2 </P>
<P> SOY UN PÁRRAFO CON <BR /> SALTO DE LÍNEA </P>
</BODY>
</HTML>
```

AÑADIMOS AL MISMO EJEMPLO COMENTARIOS, SECCIONES, PÁRRAFOS Y SALTOS DE LÍNEA. TAMBIÉN SE INCLUYE UNA LÍNEA PARA PODER VER LOS TILDES EN EL NAVEGADOR.

- **TABLE:** MARCA QUE DEFINE UNA TABLA (TODAS LAS MARCAS DE LOS ELEMENTOS DE TABLA VAN DENTRO DE ESTA).
- **TR:** MARCA QUE CONTIENE LA FILA DE UNA TABLA (CONJUNTO DE CELDAS).
- **TD:** MARCA QUE CONTIENE EL DATO DE UNA TABLA (UNA CELDA).
- **TH:** MARCA QUE CONTIENE EL ENCABEZADO DE LA FILA DE UNA TABLA (SE UTILIZAN COMO EL PRIMER CONJUNTO DE CELDAS, EN CASO DE USARSE. PUEDEN IR DESPUÉS, EL HTML LO ACEPTA, PERO NO SE VERÁ BIEN SU TABLA).

PHP HYPERTEXT PRE-PROCESSOR – PRE-PROCESADOR DE HIPERTEXTO PHP

- ▶ COMO SE PUEDE VER, POSEE UN NOMBRE RECURSIVO. ORIGINALMENTE CONOCIDO COMO PHP TOOLS – PERSONAL HOME PAGE TOOLS, "HERRAMIENTAS PARA LA PÁGINA DE INICIO PERSONAL".
- ▶ LENGUAJE DE PROGRAMACIÓN, DISEÑADO INICIALMENTE PARA ENFOCARSE EN EL DESARROLLO WEB.
- ▶ N PARTICULAR, ES EL QUE NOS PERMITIRÁ CONECTAR LAS VISUALIZACIONES DE LA PÁGINA CON LA BASE DE DATOS.
- ▶ ADEMÁS DE CÓDIGO PHP, EL ARCHIVO .PHP PUEDE CONTENER HTML, CSS Y JAVASCRIPT.
- ▶ EL CÓDIGO SE EJECUTA EN EL SERVIDOR, Y RETORNA UNA VISTA EN HTML.

A CONSIDERAR

PARA PODER CORRER PHP DE FORMA LOCAL, ES NECESARIO TENERLO DESCARGADO E INSTALADO (EL PROCESO PUEDE LLEGAR A SER BASTANTE ENGORROSO). SI AUN ASÍ QUIEREN USARLO, SE ADJUNTA UN TUTORIAL PARA **WINDOWS, MAC Y LINUX** (PUEDEN ENCONTRAR MUCHOS OTROS EN YOUTUBE, SE DEJAN ESOS COMO REFERENCIA).

NO OBSTANTE, SE ENCUENTRA INTEGRADO EN EL SERVIDOR, POR LO QUE PUEDEN IR PROBANDO DENTRO DE SU MISMA CARPETA SIN NECESIDAD DE HACER LA INSTALACIÓN LOCALMENTE.

`<?PHP CONTENIDO ?>`

- EN ESTE CASO, TODO TIENE QUE IR CONTENIDO DENTRO DE LA MARCA `<?PHP >`.
- CONTENIDO VIENE A SER EL CÓDIGO PHP A IMPLEMENTAR.
- EL MARCADOR PERMITE SEPARAR EL CÓDIGO PHP DE TODO EL CONTENIDO HTML DENTRO DEL DOCUMENTO **.PHP**.

COMANDOS COMUNES

- **ECHO**: IMPRESIÓN DE DATOS (TAMBIÉN PUEDEN USAR **PRINT**, LAS DIFERENCIAS SON **MÍNIMAS**).
- **\$**: DECLARACIÓN DE VARIABLES.
- **..**: CONCATENACIÓN DE STRINGS.
- **ARRAY(VAR1,VAR2,...,VARN)**: ARREGLO DE ELEMENTOS (PODRÍA SER ÚTIL, DADO EL CONTEXTO).
- **//**: COMENTARIO DENTRO DEL CÓDIGO DE UNA LÍNEA (**#** TAMBIÉN SIRVE).
- **/* ... */**: COMENTARIO DENTRO DEL CÓDIGO DE VARIAS LÍNEAS.

- ▶ **FOREACH:** PERMITE ITERAR SOBRE UN ARREGLO.
- ▶ **FUNCTION FUNCTION_NAME():** PERMITE CREAR UNA FUNCIÓN.
- ▶ **INCLUDE:** PERMITE IMPORTAR MÓDULOS (TAMBIÉN SIRVE **REQUIRE**, PERO SI PHP NO ENCUENTRA EL ARCHIVO, CON INCLUDE SE EJECUTA DE TODAS FORMAS).

COMANDOS COMUNES - EJEMPLO

AHORA, PROCEDEMOS A MOSTRAR EJEMPLOS DE SU USO. A PARTIR DE AQUÍ COMENZAMOS A TRABAJAR CON EL SERVIDOR ENTREGADO. EN PARTICULAR, NOS CONECTAMOS A TRAVÉS DE SUBLIME TEXT CON EL PAQUETE **SFTP** DENTRO DEL USUARIO AYUDANTÍA. (EN LA PARTE FINAL SE INCLUYE UN PASO A PASO DE SU USO.

[HTTPS://WBOND.NET/SUBLIME_PACKAGES/SFTP/USAGE](https://wbond.net/sublime_packages/sftp/usage)

CONEXIÓN CON LA BASE DE DATOS

CONEXIÓN CON LA BASE DE DATOS

- ▶ PROBABLEMENTE, LA PARTE MÁS IMPORTANTE. NOS PERMITIRÁ ACCEDER A NUESTRA BASE DE DATOS PARA OBTENER LA INFORMACIÓN DE ESTA A TRAVÉS DE CONSULTAS.
- ▶ VEREMOS EL COMANDO PRINCIPAL PARA LA CONEXIÓN (PDO) Y LA NOMENCLATURA PARA USAR CONSULTAS A TRAVÉS DE UN EJEMPLO.

CONEXIÓN BASE DE DATOS

PARA CONECTARSE SE DEBE CREAR UNA INSTANCIA DE **PDO** Y PASARLE LOS PARÁMETROS:

- DATABASE NAME
- HOST
- PORT
- USER
- PASSWORD

PDO ? -> PHP DATA OBJECT ([HTTP://PHP.NET/MANUAL/EN/BOOK.PDO.PHP](http://php.net/manual/en/book.pdo.php))

CONEXIÓN CON BASE DE DATOS

```
$DB = NEW PDO( "PGSQL:DBNAME=GRUPOXX;  
              HOST=LOCALHOST;  
              PORT=5432;  
              USER=GRUPOXX;  
              PASSWORD=X );  
$QUERY = "SELECT ....";  
$RESULT = $DB -> PREPARER($QUERY); $RESULT -> EXECUTE();  
$DATA = $RESULT -> FETCHALL();
```

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="TEXT/HTML; CHARSET=UTF-8" /> <TITLE>EJEMPLO 6</TITLE>
</HEAD>
<BODY>
<?PHP
# USAMOS TRY-CATCH POR SI FALLA LA CONEXIÓN.
TRY{
    $DB = NEW PDO("PGSQL:DBNAME=GRUPO40;HOST=LOCALHOST;USER=AYUDANTIA ;PASSWORD=XXXXXX"); }
# SI HAY ERROR, IMPRIMIR MENSAJE DE ERROR.

CATCH(PDOEXCEPTION $E){ ECHO $
E -> GETMESSAGE();
}
# CONSULTA A PROBAR.
$query = "SELECT * FROM AYUDANTES;";
$result = $DB -> PREPARE($query);
$result -> EXECUTE();

# OJO, FETCH() NOS RETORNA LA PRIMER FILA, FETCHALL() # RETORNA TODAS.
$AYUDANTES = $result -> FETCHALL();
ECHO "<TABLE>. <TR>. <TH>NOMBRE</TH>. <TH>APELLIDO</TH>. </TR>";
FOREACH ($AYUDANTES AS $AYUDANTE) {
ECHO "<TR><TD>$AYUDANTE[0]</TD><TD>$AYUDANTE[1]</TD></TR>"; }
ECHO "</TABLE>"; ?>
</BODY>
</HTML>
```

BONUS:

- BOOTSTRAP
- CSS
- JAVASCRIPT

DE DÓNDE ESTUDIAR:

- GOOGLE
- CODECADEMY
- W3SCHOOLS (**RECOMENDADO**)