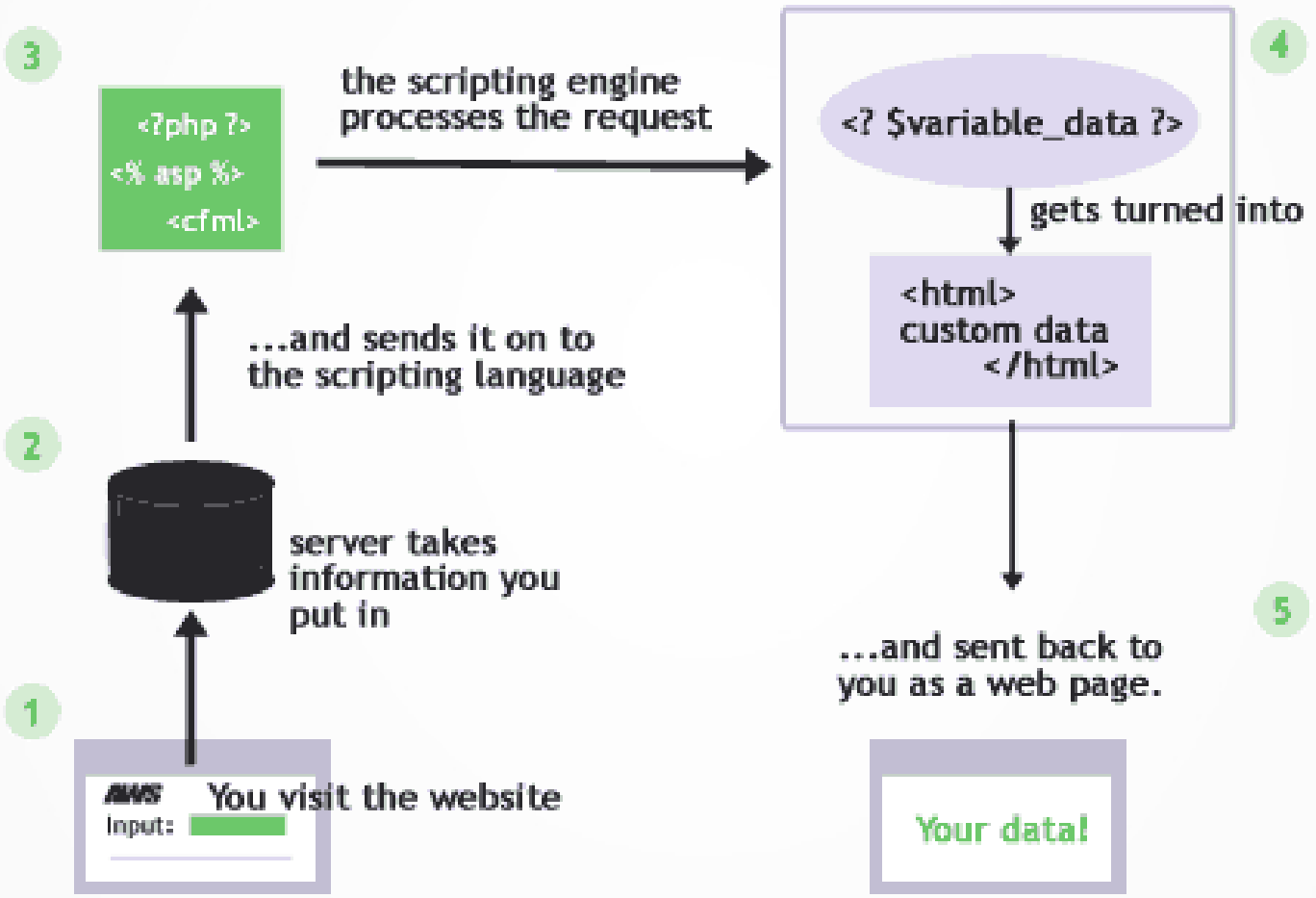
“Introducción a PHP como lenguaje Server-side”

Lenguaje Server-Side

- Son aquellos que se procesan en el servidor y retornar un resultado al cliente, estos podemos citar los siguientes:
  - PHP
  - ASP
  - JAVA (JSP, Servlets)
  - Python
  - Ruby

Lenguaje Server-Side



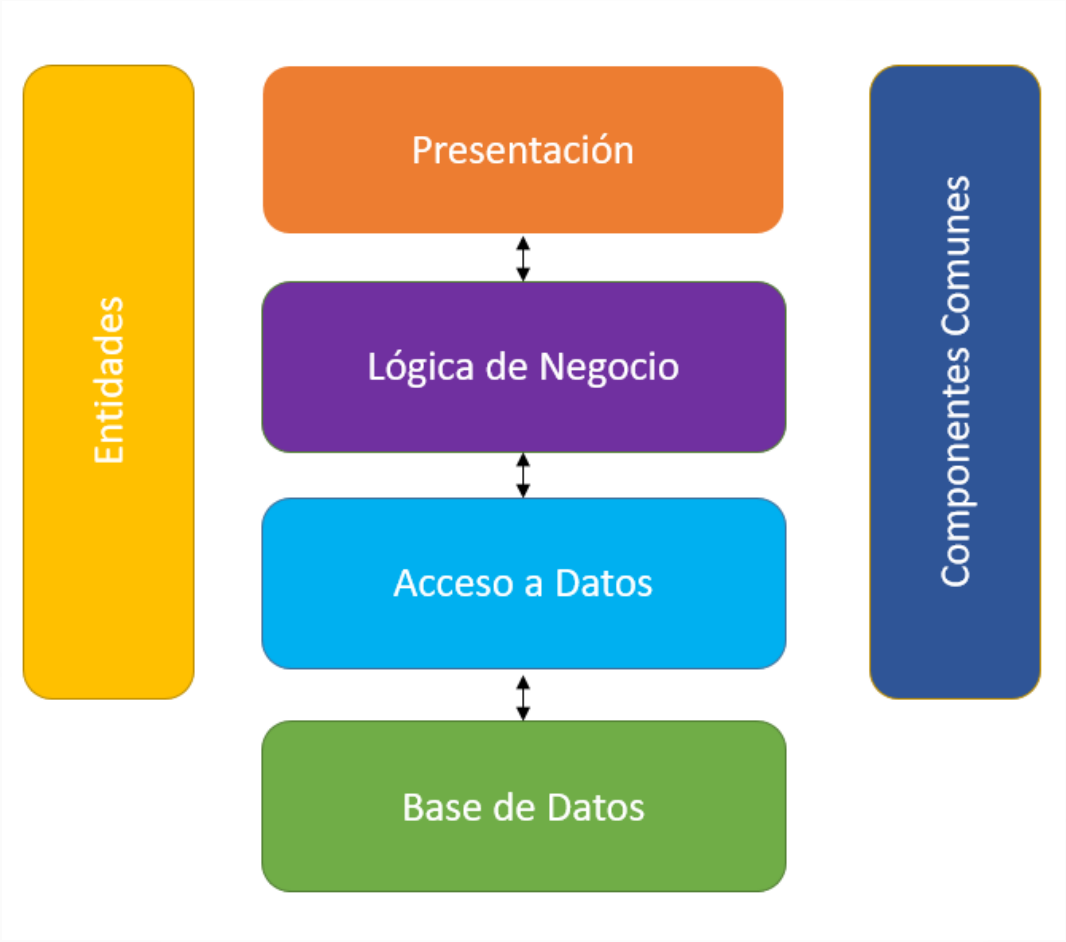
PHP

- Es un lenguaje de programación interpretado que se ejecuta del lado del servidor.
- Su función principal es retornar una página web como resultado.
- Sintaxis similar al lenguaje C y todos los derivados del mismo como JavaScript
- Como cualquier lenguaje posee estructuras de control, ciclos repetitivos, funciones, clases, librerías, etc.

Servidores de Prueba

- La mayoría de los hosting que usan PHP usan el sistema operativo Linux.
- Servidor en Linux
  - Apache
  - PHP5
  - MySQL/PosgresSQL
- Servidor en Windows:
  - WAMP o WAPP

Estructura en Capas



Estructura en Capas

La Arquitectura N capas es un modelo de desarrollo software que nos permite separar las partes que componen un sistema.

Capa Entidades

Esta capa se encarga de almacenar la entidades del negocio, es decir, guardar la información que traemos o enviamos a la base de datos. Las Entidades que manejaremos para el sistema de gestión de citas corresponden a las tablas que creamos en la BD (Paciente, Medico, Especialidad, etc.).

Capa Componentes Comunes

En esta capa van las clases que sirven de ayuda o que tienen un uso genérico dentro de nuestro proyecto, como por ejemplo: clases para cifrar y descifrar cadenas, clases con métodos de extensión, clases para validar o parsear datos entre otros.

Estructura en Capas

Capa Acceso a Datos

En esta capa van todos los métodos que sirven para traer o enviar datos a la Base de Datos (Grabar, Modificar, Eliminar, Listar, etc.).

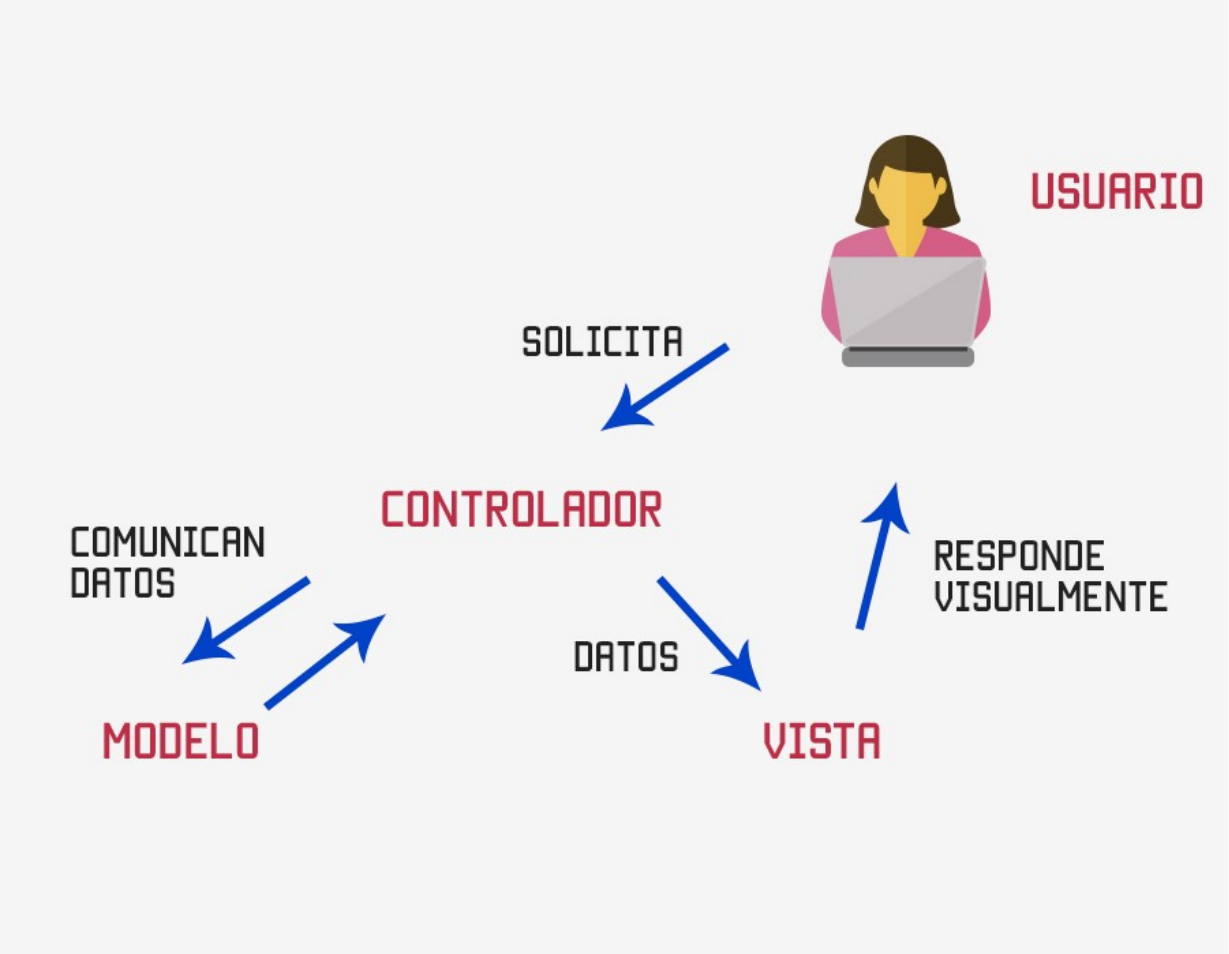
Capa Lógica de Negocio

En esta capa van las validaciones del negocio, por ejemplo: al momento de registrar una cita, validar que la cita se registre dentro del rango de horas en la que atenderá el médico (horario).

Capa Presentación

Esta capa es la que se utilizan los clientes u otros sistemas para comunicarse con nuestra aplicación

MVC


USUARIO
SOLICITA
CONTROLADOR
COMUNICAN
DATOS
RESPONDE
VISUALMENTE
DATOS
MODELO
VISTA

MVC

Modelo

Se encarga de los datos, generalmente (pero no obligatoriamente) consultando la base de datos. Actualizaciones, consultas, búsquedas, etc. todo eso va aquí, en el modelo.

Controlador

Se encarga de... controlar, recibe las órdenes del usuario y se encarga de solicitar los datos al modelo y de comunicárselos a la vista.

Vistas

Son la representación visual de los datos, todo lo que tenga que ver con la interfaz gráfica va aquí. Ni el modelo ni el controlador se preocupan de cómo se verán los datos, esa responsabilidad es únicamente de la vista.

PHP y HTML

```
<html>
    <head>
        <title>Titulo del documento</title>
    </head>
    <body>
        <p>ESTOY EN EL HTML, ANTES DEL CÓDIGO PHP</p>
        <?php
            ESTE BLOQUE DE CÓDIGO ES PHP, DEBE RESPETAR
        LA  SINTAXIS  DEL LENGUAJE PARA NO DEVOLVER
        UN MENSAJE DE  ERROR.
         ?>
    </body>
</html>
```

Inclusión

- Se pueden incluir el contenido de un archivo PHP dentro de otro PHP antes de que el mismo se ejecute.
- Existen dos funciones que se pueden usar para esto, include() y require().
- Esto es una gran ayuda al desarrollador porque permite agregar funciones, librerías, headers, footers o elementos para reutilizar en distintas páginas.
- Así de esta manera para modificar algo no se debe modificar varios archivos.

include()

- La función include() incluye todo el texto del archivo indicado dentro del archivo en el cual se usa la función.
- Si hay algún problema se genera un warning y el script continua.

- Archivo menu.php
  - Inicio -
  - Producto -
  - Empresa -
  - Contacto

- Archivo test.php

  - <html>
    - <body>
      - <?php include("menu.php"); ?>
      - Ejemplo de como incluir un archivo!
    - </body>
  - </html>

require()

- La función require() incluye todo el texto del archivo indicado dentro del archivo en el cual se usa la función.
- Si hay algún problema se genera un fatal error y el script se detiene.

- Archivo menu.php
  - Inicio -
  - Producto -
  - Empresa -
  - Contacto

- Archivo test.php

  - <html>
    - <body>
      - <?php require("menu.php"); ?>
      - Ejemplo de como incluir un archivo!
    - </body>
  - </html>

require_once(), include_once()

- Las funciones require_once e include_once son idénticas a require e include.
- La diferencia principal es que se verifica si el archivo ya fue incluido antes y no lo vuelve a ingresar.

- echo.php

```
<?php
	echo "Hola";
?>
```

- test.php

```
<?php
	require('echo.php');
	require_once('echo.php');
?>
```

Control de versiones

- Se llama control de versiones a la gestión de los cambios sobre los elementos de algún producto o una configuración
- Esto puede ser hecho de forma manual, pero es conveniente tener una herramienta para esta función
- Algunas herramientas utilizadas son
  - CVS
  - Subversion
  - SourceSafe
  - ClarCase
  - Bazzar
  - GIT
  - Mercurial
  - Etc.

Control de versiones

- Características
  - Mecanismo de almacenamiento
  - Posibilidad de realizar cambios sobre los elementos almacenados
  - Registro histórico de las acciones realizadas.
- Terminología
  - Repositorio:
    - Lugar donde se almacenan los datos actualizados e históricos de cambios
  - Revisión
    - Es una versión determinada de la información que se gestiona. Algunos sistemas lo identifican con un contador (Subversion) y otros con un código hash (GIT SHA1)
  - Tag
    - Es un nombre o etiqueta que se da a una revisión
  - Línea Base
    - Revisión aprobada de un documento o código fuente
  - Branch
    - Instante de tiempo en que se despende una copia y se evoluciona de forma independiente a la línea base

Control de versiones

- Terminología
  - Checkout
    - Es el despliegue de una versión especifica localmente de una revisión
  - Commit
    - Cuando se actualiza una revisión con los datos locales.
  - Conflicto
    - Ocurre cuando el sistema no puede manejar los cambios realizados por dos o mas usuarios en un mismo archivo.
  - Resolver
    - Acto del usuario para solucionar un conflicto
  - Cambio
    - Representa una modificación especifica sobre un archivo
  - Lista de cambios
    - Historial de los cambios realizados son los archivos.
  - Merge
    - Fusión de dos conjuntos de cambios sobre un fichero o un conjunto de ficheros en una revisión unificada

Control de versiones

- Arquitecturas
  - Distribuidos
    - Cada usuario tiene su propio repositorio. Los distintos repositorios pueden intercambiar y mezclar revisiones entre ellos.
    - Ventajas
      - Necesita menos veces estar conectado a la red para hacer operaciones
      - Aunque se caiga el repositorio remoto la gente puede seguir trabajando
      - Al tener la información replicada localmente, se necesita menos mecanismo de recuperación ante perdidas del sistema central.
      - El servidor remoto requiere menos recursos que los que necesita un servidor centralizado
      - Al ser sistemas nuevos son mas fáciles de trabajar con ramas.
  - Centralizados
    - Repositorio centralizado de todo el código, del cual es responsable un único usuario.
    - Ventajas
      - Mayor control al trabajar en equipos
      - Las versiones vienen identificadas por un numero de versión.

Control de versiones