En esta oportunidad quiero explicar varios conceptos mientras desarrollo una aplicación para la creación de encuestas o exámenes en línea. En cada tutorial separado explicare más detalladamente cada uno de los puntos de esta entrada, ¿Que vamos a aprender mientras creamos esta aplicación? Muy interesante, voy a enseñarles todo lo que yo utilizo a la hora de crear una página web:

A la hora de crear la página vamos a tener las siguientes premisas fundamentales:

* **Pensar todo el tiempo en el usuario:** Vamos a evitar cualquier actividad extra para el usuario, crear ayudas en todo momento, prevenir en lugar de corregir, darle velocidad a la pagina, interfaz mimalista, etc.
* **La simplicidad no tiene nada que ver con el tamaño:** Tu puedes tener la pagina más grande del mundo pero que siga siendo muy simple desde todo punto de vista (arquitectura, organización, usabilidad, etc.)
* **Estandarizar los procesos:** Hay actividades que son muy repetitivas, hay que tratar de estandarizarlas o incluso tratar de automatizarlas para que cada vez se hagan mas rápido y sin errores.
* **Patrones de diseño:** Todos los problemas que se nos puedan presentar a la hora de crear una página web ¡ya se le presentaron a otra persona!, busca en internet, utiliza los patrones de diseño, tal vez trabajemos mas al principio pero luego estaremos volando y sin equivocaciones.

1. **¿Cómo diseñar una página con maquetación CSS?**

Primero vamos a maquetar la página en contenedores <div> para utilizar estándares de CSS, así vamos a poder diseñar una página pensando en la información y el contenido y luego le daremos un diseño creando hojas de estilos que tendrán clases que se les aplicaran a los contenedores que definimos, algo asi:

Fíjense que la pagina web html solo contiene contenedores de información que no tienen ningún diseño (la cabecera de la pagina, el contenido y el pie de la pagina), cuando se le aplica la capa de la derecha (la hoja de estilos) los contenedores cambian de color, tipo de letra, color de letra, sangrías para el texto, etc. Cuando nosotros queramos podemos cambiar la hoja de estilos y nuestra aplicación se vera muy diferente pero seguirá teniendo la misma estructura e información.

1. **¿Cómo utilizar plantillas para poder compartir la misma cabecera, menú y pie en todas las paginas sin tener que copiar y pegar el código?**

La idea es crear una archivo “header.php” que será la cabecera de las paginas y lo importaremos cada vez que queramos renderizar o mostrar una página en especifico, de esta manera si realizamos un cambio en esta cabecera no tendremos que ir pagina por pagina realizando el cambio en todas las cabeceras sino que cambiaríamos el archivo header.php y el cambio se reflejaría en todas las paginas. Lo mismo ocurriría en el caso del menú (menu.php) y el pie de la pagina (footer.php).

1. **¿Cómo evitar recargar la pagina tan amenudo?**

Con las nuevas tecnologías es posible recargar contenedores específicos de las páginas web en vez de recargar toda la pagina, de esta manera aumentamos mucha facilidad de uso y tiempo de espera. Además recargamos menos al servidor ya que realizamos menos peticiones y por ultimo vamos a ver cómo termina siendo incuso ¡Mucho más fácil que antes! Trabajar de esta manera. Nosotros utilizaremos el framework de javascript llamado JQuery para lograr estas funcionalidades.

1. **¿Cómo estructurar nuestra pagina de una manera sencilla y organizada para que siga siendo simple a pesar de que ser bastante grande?**

Luego de buscar y probar la gran mayoría de los frameworks de desarrollo yo decidí crear mi propia arquitectura para mis páginas web, siguiendo algunos estándares y patrones de diseño pero sin volverla tan compleja y abstracta como me parece que ocurre en el caso de frameworks existentes como symfony.

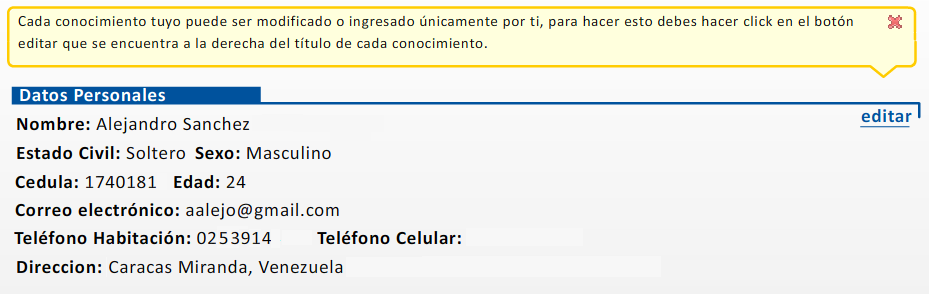
1. **¿Cómo dividir en capas toda la pagina web para lograr la independencia de cada una de ellas?**

En la arquitectura que diseñe la pagina web se divide en 6 capas donde cada una de ellas es completamente independiente, es decir, si quieres cambiar de base de datos solo tienes que cambiar la capa de comandos, si quieres cambiar el diseño de la pagina, solo tienes que cambiar la capa de diseño, etc.

En mi arquitectura yo planteo las siguientes capas:

1. Diseño: Renderiza los datos, les da colores, tamaño, etc. Puras hojas de estilo.
2. Vista: La estructura de los datos, contiene la información y no contienen practicamente ninguna lógica.
3. Cliente: Creación de componentes DatePicker, Slider, etc. Validaciones, Evitar recargo de la pagina, ayuda de usuario, etc.
4. Lógica: Transacciones (crudManager), Plantillas (templateManager), Permisos de usuario, Validaciones, etc.
5. Comando: Enlace entre el modelo y la lógica para independizar a la página de la estructura de los datos.
6. Modelo: Puede ser una BD, unos web services, o cualquier otra forma de obtener y guardar datos.
7. **¿Cómo crear una ayuda de usuario sobre la pagina?**

El problema con las ayudas es que siempre son una sección aparte de la aplicación y el usuario ni siquiera sabe cómo encontrar la ayuda para lo que está buscando. Otra de las ventajas que te da estructurar tu página en contenedores <div> es que tienes la posibilidad de mostrarlos y ocultarlos fácilmente. De esta manera si nosotros consideramos que la pagina actual que estamos creando debemos colocar un espacio de ayuda entonces podemos crear un contenedor con esa información y en cualquier momento el usuario podrá ocultarlo y continuar como si nada. Por ejemplo:



1. **¿Cómo prevenir en lugar de corregir?**

La parte más importante a la hora de crear una web es como se plantea la interacción del usuario con la aplicación, la mayoría de las paginas utilizan los componentes estándares de formulario que tiene html para lograr esta interacción, pero lamentablemente estos componentes son muy limitados y en un gran porcentaje de las veces dificultan el uso de la pagina. Para eso en la arquitectura que estamos planteando hay un espacio dedicado a la creación o importación de nuevos componentes javascript. ¿Por qué introducir una fecha en un TextInput en una seria de ComboBoxs? Lo más natural seria un calendario, ¿no les parece? Existen una serie de componentes como NumericStepper, Slider, DatePicker, etc. Que hacen que la experiencia de usuario se vuelva mucho más simple y práctica y además evitan muchísimas validaciones hacia nosotros porque no dejan espacio para los errores. Y si además colocas tremenda ayuda, ¡es excelente!

1. **¿Cómo hacer una página segura?**

Lo primero que debemos tomar en cuenta a la hora de hacer una página segura es que la capa de javascript no puede ser tomada en cuenta para la seguridad en casi nada, ¿Por qué? Porque puede ser desactivada en cualquier momento por los hackers, incluso los navegadores más viejos insiquiera soportan javascript. ¿Qué quiere decir esto? Pues muy fácil, las validaciones de datos que hacemos en javascript tambien las debemos hacer en PHP o en cualquier lenguaje que estemos usando del lado del servidor. Pero OJO: Igual debemos seguir haciendo las validaciones del lado del cliente para no recargar al servidor y para seguir con nuestra premisa de PREVENIR en lugar de CORREGIR.

Para hacer nuestra arquitectura segura yo decidí dividir toda la interacción con los datos en transacciones, si se fijan en la capa lógica existe el crudManager y el infoManager. El primero de estos se encarga de eliminar, modificar y insertar los datos. El segundo se encarga solo de consultaros, por lo tanto requieren permisos de usuario diferentes para ser utilizados. Los mismo ocurre con el templateManager que se encarga de armar las vistas que el usuario va a visualizar, si el usuario no tiene permisos para visualizarlas entonces lo lleva a una página de error.

¿Cómo se cargan los permisos? En variables de sesión, en la arquitectura que vamos a utilizar ya existe un modulo de autenticación que solo debemos realizarle pequeños cambios para adaptarlo a nuestra página, al momento de iniciar sesión te otorga permisos de usuario que luego tu controlas para acceder a las vistas y transacciones desde el crudManager, infoManager y templateManager.

1. **¿Cómo realizar los crud sin recargar la pagina?**

Es demasiado común en una página web querer insertar, modificar, eliminar y consultar valores de una entidad en especifico (Persona, Paciente, Profesor, etc.). Con unos pocos conceptos es posible estandarizar estos procedimientos para hacerlos más fáciles de programar y además se logra a la misma vez evitar recargar la pagina tan a menudo. Si estamos utilizando la arquitectura CANIS (De la que he hablado durante toda esta entrada) solo debemos crear dos contenedores, uno para visualizar la información y otro para modificarla, por ejemplo: En el siguiente ejemplo se tiene una lista de anuncios publicada por un condominio, y se quiere modificar un anuncio en particular:

Este sería el código del contenedor para visualizar los datos:

<div id="showPost1" class="post">  
 <h1 class="title"><a href="#">Racionamiento de agua</h1>  
 <p>Los dias lunes y martes no entrará agua al edficicio de la calle, por lo tanto se racionará el agua del tanque en el siguiente horario: Lunes: de 8am a 12pm - de 2pm a 8pm. | Martes: de 8am a 12pm - de 2pm a 8pm.</p>  
 <p style="text-align:right;">  
 <a id="detallesLink" class="editLink" name="Post1>" href="#">editar</a>   
 <a class="deleteLink" name="Post" href="crudManager.php?action=delete\_anuncio&codigo=1">Eliminar</a>  
 </p>  
</div>

Y este sería el contenedor para modificar los datos:

<div id="editPost1" class="post" style="display:none;">  
 <form action="crudManager.php" method="post" id="anuncioForm">  
 <input name="action" type="hidden" value="edit\_anuncio" />  
 <input name="codigo" type="hidden" value="1" />  
 <h1 class="title">Titulo: <input name="titulo" type="text" value=” Racionamiento de agua” /></h1>  
 <textarea name="mensaje" cols="" rows="" style="width:500px; height:100px;"> Los dias lunes y martes no entrará agua al edficicio de la calle, por lo tanto se racionará el agua del tanque en el siguiente horario: Lunes: de 8am a 12pm - de 2pm a 8pm. | Martes: de 8am a 12pm - de 2pm a 8pm.</textarea>  
 <p style="text-align:right;"><input name="" type="submit" value="Guardar" /></p>  
 </form>  
</div>

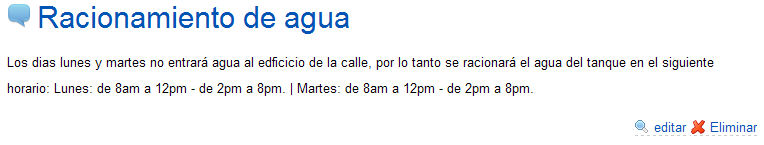
He marcado de colores las zonas que a son importantes en ambos contenedores, si estamos trabajando con CANIS, debemos decirle cual contenedor es el que se utiliza para mostrar la información y cual contenedor es el que se utiliza para editar la información, esto lo hacemos con el campo ID de cada uno de los contenedores, colocando la palabra “show” o “edit” (segun el caso) seguida de el nombre de la entidad que estamos modificando en este caso “Post” seguida de el codigo de la entidad especifica que estamos modificando (can caso de que estemos en una lista de items de la misma entidad).

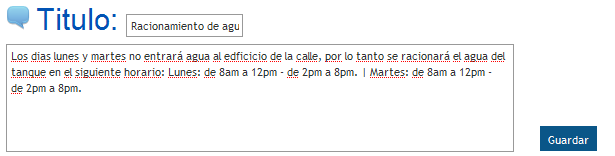
Siempre el contenedor que se creo para modificar la entidad debe tener por defecto en sus estilos la propiedad **display:none** para que la pagina se desempeñe mucho mejor, esta propiedad le dice al contenedor que por defecto el debe estar oculto.

Luego debemos irnos al vinculo que hemos definido que se le debe hacer click para modificar la entidad (ocultar el contenedor de visualizar y mostrar el contenedor de modificar) y colocarle en su propiedad class el valor “editLink”, y en su propiedad name el nombre de la entidad y código que este vinculo modifica (en este caso “Post1”) de esta manera CANIS sabrá lo que tiene que hacer.

Por último debemos irnos al vinculo que hemos definido que se le debe hacer click para eliminar la entidad y colocarle en su propiedad class el valor “deleteLink” y de esta manera CANIS sabrá lo que tiene que hacer.

Al final los dos contenedores se verían de la siguiente manera, por supuesto que no se verían al mismo momento, sino que al presionar “editar” se ocultaría el primer contenedor y se mostraría el segundo contenedor:





Bueno en las siguientes entradas explicare mucho mejor cada uno de los puntos de los que he hablado en esta oportunidad ya que todavía nos queda muuuuuucho de que hablar y buenas prácticas que seguir para conseguir mucha rapidez, de todas maneras para los que saben leer ingles aquí les dejo un pequeño documento donde explica un poco más detallada la arquitectura CANIS.