

Informe de rol Ariel Andia Roa.

Las actividades realizadas esta semana fueron:

Definir puntos a considerar para el momento de programar cada sprint que cumpla con requerimientos funcionales y el desarrollo de la interfaz gráfica de usuario, esto con el fin de minimizar los errores al momento de las pruebas. Los puntos fueron los siguientes:

Para los encargados de cada requerimiento funcional, deben tener en cuenta los siguientes puntos al momento de desarrollar su módulo del sistema.

RF_01 login.

- Controlar que los campos sean obligatorio tanto para la vista (input con “required”), como en el controlador.
- Al crear el login la contraseña debe tener mínimo 5 caracteres.
- Las claves deben ser encriptadas.
- Redireccionar al inicio (petición de claves) cuando se accede al sistema a través url a métodos cuando no hay usuario logeado.

RF_06 preinscribir oferta práctica profesional.

- Fijar fecha de vencimiento de la preinscripción y una vez pasada esa fecha eliminar la preinscripción.

RF_07 Registrar oferta práctica profesional.

- Controlar que los campos obligatorios a llenar en el formulario sean validados en el controlador, como también en la vista (input con “required”).
- Fijar fecha de caducidad de la oferta de práctica, pasada esa fecha el sistema debe realizar acciones.
- Cuidar integridad de fechas (ej: evitar que se ingrese una fecha de práctica en periodos pasados).
- Controlar el ingreso de fechas (puede ser a través del uso de datepicker de la librería JQuery UI).

Interfaz Gráfica de Usuario.

- Desarrollar diseño responsivo.
- Confirmar que los elementos usados funcionen en los navegadores más usados.
- Destacar en el código de la vista las secciones en las que se debe programar.

También se formalizó un estándar para el grupo de la programación PHP.

Estándares de programación PHP.

Indentación: Serán compuestas por 4 espacios y no por tabulación.

Tamaño máximo de líneas: La longitud recomendable para una línea de código es de 80 caracteres. Esto quiere decir que el desarrollador al programar debe intentar mantener su línea de código por debajo de los 80 caracteres, aunque se puede sobrepasar siempre y cuando sea necesaria. El tamaño máximo de cualquier línea de código PHP es de 120 caracteres.

Clases: Los nombres de clases pueden contener sólo caracteres alfanuméricos. Los números están permitidos en los nombres de clase, pero desaconsejados en la mayoría de casos. Para la notación se utilizará camelCase.

Nombre de archivos: Los nombres de los archivos no deben llevar espacios.

Funciones y Métodos: Los nombres de funciones pueden contener únicamente caracteres alfanuméricos. Los guiones bajos (_) no están permitidos. Los números están permitidos en los nombres de función pero no se aconseja en la mayoría de los casos.

Los nombres de funciones deben empezar siempre con una letra minúscula. Cuando un nombre de función consiste en más de una palabra, la primera letra de cada nueva palabra debe estar en mayúsculas.

Los nombres de función deben ser lo suficientemente elocuentes como para describir su propósito y comportamiento.

Para la programación orientada a objetos, los métodos de acceso para las instancias o variables estáticas deben ir antepuestos con un "get" o un "set".

Variable: Los nombres de variables pueden contener caracteres alfanuméricos. Las barras bajas (_) no están permitidas. Los números están permitidos en los nombres de variable pero no se aconseja en la mayoría de los casos.

Demarcación de código PHP: No se permite el uso de etiquetas cortar.

Array asociativos: Al declarar arrays asociativos con la construcción array, se recomienda partir la declaración en múltiples líneas. En este caso, cada línea sucesiva debe ser tabuladas con cuatro espacios de forma que tanto las llaves como los valores están alineados:

```
$sampleArray = array('firstKey' => 'firstValue',  
                     'secondKey' => 'secondValue');
```

También se permite hacer un salto de línea para el primer elemento del array:

```
$sampleArray = array(  
    'firstKey' => 'firstValue',  
    'secondKey' => 'secondValue'
```

```
);
```

Declaración de clases:

Las Clases deben ser nombradas de acuerdo a las convenciones definidas anteriormente. La llave "{" deberá escribirse siempre en la línea debajo del nombre de la clase ("one true brace"). Todo el código contenido en una clase debe ser separado con cuatro espacios. Únicamente una clase está permitida por archivo PHP.

Las clases que extiendan otras clases o interfaces deberían declarar sus dependencias en la misma línea siempre que sea posible.

```
class SampleClass extends FooAbstract implements BarInterface
{
}
```

Si como resultado de esas declaraciones, la longitud de la línea excede la longitud del Tamaño máximo de línea , se debe romper la línea antes de la palabra clave "extends" y / o "implements" e indentarlo con un nivel de indentación (4 espacios).

```
class SampleClass
    extends FooAbstract
    implements BarInterface
{
```