

# Proyecto integrador - Fase 02

- 1) Formulario de Contacto
- Explorar el código fuente del módulo contacto.php. Notar que el formulario contiene los campos Nombre, E-Mail, Mensaje. Identificar los atributos name de cada uno, ya que serán usados para armar el archivo de envío del e-mail de contacto.
- Crear un nuevo archivo llamado enviar.php, luego en el formulario de contacto.php actualizar el atributo action para que apunte a este nuevo archivo creado.
- En enviar.php lo primero que debe hacerse es validar que efectivamente sea invocado por el formulario, es decir corroborando que sea mediante una petición de tipo POST. Sino, debe hacerse una redirección hacia la sección de contacto (./?page=contacto).
- En caso que la petición sea POST, debe empezar el proceso de validación de los campos antes de intentar construir y enviar el e-mail de contacto. Validar de acuerdo a los siguientes criterios:
- Nombre:
  - No debe estar vacío
  - No debe ser menor a 3 letras
  - No debe ser un número
  - El primer carácter no debe ser un número
- E-Mail:
  - No debe estar vacío
  - Debe tener formato de e-mail (conteniendo un carácter "@" y un carácter de ".")
- Mensaje:
  - No debe estar vacío
  - No debe tener más de 400 caracteres
- En caso de que algún campo no valide según lo solicitado, mostrar provisoriamente un simple mensaje que diga "{Nombre | E-Mail | Mensaje} Invalido".
- Si todos los campos validan, habrá que construir el email de contacto cuya estructura sea la siguiente:

---

## ComercioIT - Datos de contacto

**Nombre:** Juan Perez

**E-Mail:** juanperez@email.com

**Mensaje:**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

---

- Para lograrlo, definir en variables cada una de las partes que compondrán el email: \$destinatario, \$asunto, \$cuerpo, \$cabeceras.
- Una vez construidas, proceder a realizar el envío con una validación en simultáneo. Si el envío es correcto, mostrar provisoriamente un simple mensaje que diga: "Consulta enviada". Si falla el envío, que el mensaje diga: "Consulta no enviada".

Nota: recordar que la función mail() puede no funcionar desde localhost. Solo cuando el proyecto esté subido a un webhosts funcionará correctamente.

- 2) Adaptando la navegación
  - Existe la posibilidad que se ingrese a la web sin especificar ningún parámetro page Por ende, conviene colocar la asignación del valor el parámetro \$\_GET["page"] a la variable \$page solo si efectivamente existe dicho parámetro. Sino, asignar directamente a \$page el valor "inicio" .
  - Crear un archivo llamado functions.php, el cual será usado para definir aquellas funciones que representen acciones concretas del funcionamiento de la web. Crear en el mismo una función llamada CargarPagina, que reciba un parámetro llamado \$pagina.
  - Colocar dentro de la función CargarPagina el include usado dentro de section#page y actualizarlo para que coincida con el parámetro que ingresa a dicha función.
  - Mediante un require\_once se incorporará el archivo functions.php al index.php, permitiendo usar la función CargarPagina en section#page en reemplazo del include usado anteriormente.
  - Probar el correcto funcionamiento de los links/url para corroborar la correcta carga de los módulos de cada sección.
- 3) Respuestas del servidor personalizadas
  - Se crearán mensajes del servidor con códigos personalizados para indicarle al usuario lo ocurrido en el proceso del envío del email de contacto. La tabla de códigos será:
    - 0x001 = Nombre inválido
    - 0x002 = E-Mail inválido
    - 0x003 = Mensaje inválido
    - 0x004 = Consulta enviada
    - 0x005 = Consulta no enviada
  - En reemplazo de los simples mensajes usados en las validaciones de cada campo, colocar redirecciones al módulo contacto, indicando adicionalmente un parámetro llamado rta con el código acorde a su mensaje. Ej: "Nombre inválido" => ./?page=contacto&rta=0x001
  - En el archivo functions.php crear una función llamada MostrarMensaje que reciba el código enviado en el parámetro rta. La misma retornará según el código recibido un mensaje más detallado y en formato HTML. Ej:

```
<p class="rta rta-0x001">El nombre ingresado no es válido</p>
```

- En el módulo contacto.php validar si existe el parámetro rta, en cuyo caso invocar a la función MostrarMensaje enviando el valor de dicho parámetro.
- Probar nuevamente el funcionamiento del formulario para corroborar las validaciones y los respectivos mensajes de cada acción.