



## PHP & Forms

---

Noviembre 11, 2019

### Resumen

En esta proyecto, trabajaremos con PHP y HTML. La idea es entender cómo podemos enviar información y files desde el cliente (navegador) al servidor (PHP), utilizando formularios.

### Contenidos

---

**Requisitos del Proyecto**

**Tareas**

**Incidencias**

**Lecciones aprendidas**

**Calendario del Proyecto**

**Cronograma del Proyecto**

**Medidas de Control de Calidad**

**Métricas de Calidad**

**Riesgos**

**Herramientas usadas**

## Requisitos del Proyecto

- Debes de usar GIT. Es importante que las indicaciones y los commits sean los suficientemente explícitos y concretos como para poder entender los cambios sin la necesidad de requerir de información adicional en la medida de los posible.
- No puedes hacer uso de Frameworks o librerías de terceros.
- Documentar todos tus algoritmos.
- Usar programación orientada a objetos.

## Tareas a realizar

1. Leer el documento descriptivo del proyecto:
  - Prioridad: Alta.
  - Descripción: Leer la descripción del proyecto para entender todos los requisitos.
  - Nivel de dificultad: 2/10
  - Estimación de tiempo: 15 min.
2. Crear repositorio en GitHub
  - Prioridad: Alta.
  - Descripción: Crear el repositorio en Github para almacenar todos los contenidos los archivos del proyecto.
  - Nivel de dificultad: 4/10
  - Estimación de tiempo: 10 min.
3. Prototipar:
  - Prioridad: Alta.
  - Descripción: Conceptualizar cómo será el quiz, hacer bosquejos de cómo será la página.
  - Nivel de dificultad: 5/10
  - Estimación de tiempo: 30 min
4. Conceptualización del proyecto
  - Prioridad: Alta
  - Descripción: crear las plantillas html a nivel general para cada página del quiz
  - Nivel de dificultad: 4/10
  - Estimación de tiempo: 60 min.
5. Descargar archivos para PHP
  - Prioridad: Alta
  - Descripción: descargar los archivos necesarios (como XAMP y Apache), para poder trabajar con php.
6. Crear HTML:
  - Prioridad: Alta.
  - Descripción: crear los html para crear los “form” y los “input” para permitir al usuario guardar los files al servidor.
  - Nivel de dificultad: 2/10.
  - Estimación de tiempo: 5 min.
7. Estudio de PHP
  - Prioridad: Media.
  - Descripción: hacer un estudio breve general sobre cómo funciona PHP.
  - Nivel de Dificultad: 5/10.
  - Estimación de tiempo: 180 min.

8. Formularios HTML

- Prioridad: Alta.
- Descripción: crear los formularios de html usando el tag <form>.
- Nivel de dificultad: 3/10.
- Estimación de tiempo: 10 min.

9. Analizar tipos de métodos HTTP:

- Prioridad: Alta.
- Descripción: analizar qué tipo de métodos HTTP puedes implementar en el uso de formularios HTML.

10. Funciones PHP

- Prioridad: Alta.
- Descripción: Escribir las funciones necesarias para el desarrollo del proyecto.
- Nivel de dificultad: 8/10.
- Estimación de tiempo: 3 o 4 horas.

11. Subir archivos

- Prioridad: Alta.
- Descripción: Poder lograr que los archivos se suban al servidor.
- Nivel de dificultad: 9/10.
- Estimación de tiempo: 3 horas.

12. Validar archivos

- Prioridad: Alta.
- Descripción: validar que los archivos sean de formato '.jpg', '.txt', etc
- Nivel de dificultad: 8/10.
- Estimación de tiempo: 2 horas.

13. Cambio de nombre de archivos

- Prioridad: Alta.
- Descripción: lograr que los archivos cambien de nombre al subirlos.
- Nivel de dificultad: 8/10.
- Estimación de tiempo: 3 horas.

14. Preparar la presentación.

- Prioridad: Alta.
- Descripción: preparar la presentación del proyecto.
- Nivel de dificultad: 6/10
- Estimación de tiempo: 3 horas.

## **Registro de Incidencias**

- Enfrente muchas dificultades a la hora de trabajar con PHP ya que era un lenguaje nuevo que no conocía.
- Me costó poder llegar a subir los archivos al servidor.
- Me faltó estudiar más sobre el funcionamiento de PHP desde el principio.
- Al inicio tuve dificultades con el manejo de estrés, ya que me sentía muy apurado por terminar el proyecto, dejando un poco al lado el entendimiento de lo que estaba haciendo. Esto afectó mucho mi desarrollo del proyecto y me hizo perder tiempo.

## Lecciones aprendidas

- A pesar de su antigüedad, PHP sigue siendo una tecnología de uso común en el desarrollo web.
- PHP está diseñado para interactuar con HTML para generar sitios web dinámicos.
- Cada declaración en PHP debe terminarse con un punto y coma;
- Los archivos PHP tienen una extensión .php y el archivo siempre comienza con la etiqueta PHP de apertura <? Php. La etiqueta de cierre está implícita y se excluye por convención.
- El espacio en blanco generalmente se ignora al ejecutar código PHP.
- Las palabras clave no distinguen entre mayúsculas y minúsculas en PHP. Como convención, use la carcasa estándar.
- Los comentarios de una sola línea se hacen en PHP usando # o //. Los comentarios de varias líneas se colocan entre /\* y \*/.
- Para comentar un código PHP, debes usar el opening multiline comment syntax.

## **Herramientas utilizadas**

Para el desarrollo de este proyecto se utilizaron las siguientes herramientas:

- Visual Studio Code: para escribir el código.
- Xdebug: debugger para trabajar con PHP.
- Git: para guardar los cambios del proyecto.
- Google y YouTube: para búsqueda de información y tutoriales.

## **Uso de Git**

Para este proyecto se utilizó únicamente el branch Master para guardar los cambios, por lo que se podría afirmar que la estrategia del uso de Git fue básica y sencilla.



## Calendario

- Miércoles
  - ❖ Preparar a nivel básico la maquetación de las páginas principales.
  - ❖ Agregarle diseño a las páginas principales, con CSS, por ejemplo márgenes.
- Jueves:
  - ❖ Trabajar en funcionalidades de la primera página de preguntas.
  - ❖ Estudiar y entender sobre Modales, para poder cambiar las preguntas.
  - ❖ Ver tutoriales de YouTube para entender sobre el localStorage.
  - ❖ Reunirme con Rafael para ver el tema de cómo crear las barras de progreso con html y css
- Viernes:
  - ❖ Trabajar con Javascript
  - ❖ Entender más cómo funciona el localStorage.
  - ❖ Intentar que el contenido del array de preguntas pueda reemplazarse al html con la propiedad innerhtml
- Sábado:
  - ❖ Seguir intentando crear funciones con javascript.
  - ❖ Mejoras generales a la organización del proyecto (creación de folders para archivar).
  - ❖ Conceptualizar las tareas a seguir.
- Domingo:
  - ❖ Intentar terminar las últimas fases del proyecto, por ejemplo poder cambiarle el nombre a un archivo que se ha subido dos veces.

### Cronograma

#Tarea	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
1	X						
2	X						
3	X						
4	X						
5		X					
6			X				
7			X				
8			X				
9			X				
10				X			
11					X		
12						X	
13							X

### **Métricas de Calidad**

Para este proyecto se utilizó el Xdebug para estar seguros de que el código estaba funcionando como debería.