



# Universidad Tecnológica de Puebla

30-5-2021

**Materia:**

Desarrollo Web Integral

**Profesor:**

Javier Nolasco Hernández

**Tarea:**

Producto 1

**Integrantes:**

Gonzalez Hernández Israel

Gonzalez Rueda Atlai

Martínez de Santiago Giovanni

Gonzalez Flores Daniel

**Cuatrimestre:**

9°

**Grupo:**

C

## Contenido

INTRODUCCION .....	1
METODOLOGÍA ÁGIL .....	2
<b>Roles en el equipo Scrum</b> .....	2
<b>Ventajas de la metodología Scrum</b> .....	3
<b>Desventajas de la metodología Scrum</b> .....	3
PLAN DEL PROCESO DE DESARROLLO WEB.....	4
JUSTIFICACIÓN Y DIAGRAMA DE ARQUITECTURA BASADA EN EL MODELO MVC.....	7
<b>Modelos</b> .....	7
<b>Vistas</b> .....	7
<b>Controladores</b> .....	7
<b>Arquitectura de aplicaciones MVC</b> .....	8
<b>Lógica de negocio / Lógica de la aplicación</b> .....	¡Error! Marcador no definido.
JUSTIFICACIÓN DEL ENTORNO DE DESARROLLO WEB A UTILIZAR, ADEMÁS DE INCLUIR LA ACTIVIDAD DE CONFIGURACIÓN.....	10

## INTRODUCCION

En este producto se documentará lo que es una metodología ágil en base a nuestro proyecto, así como también tendremos 2 justificaciones.

Una justificación se basará en modelo MVC contendrá un diagrama el cual muestre el proceso, la otra justificación es en el entorno de desarrollo web que se esta utilizando en el proyecto.

## METODOLOGÍA ÁGIL

La metodología ágil que se está utilizando es SCRUM, La metodología Scrum es un marco de trabajo o framework que se utiliza dentro de equipos que manejan proyectos complejos. Es decir, se trata de una metodología de trabajo ágil que tiene como finalidad la entrega de valor en períodos cortos de tiempo y para ello se basa en tres pilares: la **transparencia, inspección y adaptación**. Esto permite al cliente, junto con su equipo comercial, insertar el producto en el mercado pronto, rápido y empezar a obtener ventas.

Scrum se basa en aspectos como:

- ✚ La flexibilidad en la adopción de cambios y nuevos requisitos durante un proyecto complejo.
- ✚ El factor humano.
- ✚ La colaboración e interacción con el cliente.
- ✚ El desarrollo iterativo como forma de asegurar buenos resultados.

### 1. Transparencia

Con el método Scrum todos los implicados tienen conocimiento de qué ocurre en el proyecto y cómo ocurre. Esto hace que haya un entendimiento “común” del proyecto y una visión global.

### 2. Inspección

Los miembros del equipo Scrum frecuentemente inspeccionan el progreso para detectar posibles problemas. La inspección no es un examen diario, sino una forma de saber que el trabajo fluye y que el equipo funciona de manera autoorganizada.

### 3. Adaptación

Cuando hay algo que cambiar, el equipo se ajusta para conseguir el objetivo del sprint. Esta es la clave para conseguir el éxito en proyectos complejos, donde los requisitos son cambiantes o poco definidos y en donde la adaptación, la innovación, la complejidad y flexibilidad son fundamentales.

## Roles en el equipo Scrum

el equipo tiene como foco entregar valor y ofrecer resultados de calidad que permitan cumplir los objetivos de negocio del cliente. En la metodología Scrum hay tres roles muy importantes los cuales son **Product Owner**, **Scrum Master** y por último el **Equipo de Desarrollo**.

### 1. Product owner (Israel Gonzalez Hernández):

Es el responsable de maximizar el valor del trabajo del equipo de desarrollo. La maximización del valor del trabajo viene de la mano de una buena gestión del Product Backlog.

El Product owner es el único perfil que habla constantemente con el cliente, lo que le obliga a tener muchos conocimientos sobre negocio.

Para finalizar, un equipo Scrum debe tener solo un Product Owner y este puede ser parte del equipo de desarrollo.

## **2. Scrum Master (Atlai Gonzalez Rueda):**

Es el responsable de que las técnicas Scrum sean comprendidas y aplicadas en la organización. Es el manager de Scrum, un líder que se encarga de eliminar impedimentos o inconvenientes que tenga el equipo dentro de un sprint (que ya revisaremos en detalle más adelante), aplicando las mejores técnicas para fortalecer el equipo de marketing digital.

Dentro de la organización, el Scrum Master tiene la labor de ayudar en la adopción de esta metodología en todos los equipos.

## **3. Equipo de desarrollo (Giovanni Martínez de Santiago y Daniel Gonzalez Flores):**

Son los encargados de realizar las tareas priorizadas por el Product Owner. Es un equipo multifuncional y autoorganizado. Son los únicos que estiman las tareas del product backlog, sin dejarse influenciar por nadie.

Los equipos de desarrollo no tienen sub equipos o especialistas. La finalidad de esto es transmitir la responsabilidad compartida si no se llegan a realizar todas las tareas de un sprint.

### **Ventajas de la metodología Scrum**

Scrum es muy fácil de aprender: los roles, hitos y herramientas son claros y tienen un objetivo por lo que es un método muy relacionado con nuestra manera diaria de trabajar.

El cliente puede comenzar a usar el producto rápidamente.

Se agiliza el proceso, ya que la entrega de valor es muy frecuente.

Menor probabilidad de sorpresas o imprevistos, porque el cliente está viendo frecuentemente el proyecto.

### **Desventajas de la metodología Scrum**

Aunque Scrum sea fácil de aprender, es muy difícil implementarlo. Esto supone una predisposición y un cambio de cultura de la organización que debe ir desde los altos mandos hasta los clientes.

La necesidad de tener equipos multidisciplinarios puede ser un problema, ya que es difícil encontrar personas que sean capaces de hacer todo el trabajo de un equipo.

El equipo puede tender a realizar el camino más corto para conseguir el objetivo de un sprint, el cual no siempre ofrece resultados de calidad.

#### PLAN DEL PROCESO DE DESARROLLO WEB

Nuestro plan del proceso para el desarrollo web es el siguiente:

En primera instancia nosotros elegiremos como va a quedar conformado nuestro equipo de desarrollo lo cual debemos de elegir los roles en base a nuestra metodología SCRUM de cada uno de nuestros integrantes para poder llevar acabó los siguientes pasos, los roles quedaron de la siguiente manera:

- 🚦 Israel Gonzalez Hernández se encargará del rol de “Product Owner”
- 🚦 Atlai Gonzalez Rueda será el encargado del rol de “Scrum Master”
- 🚦 Giovanni Martínez de Santiago y Daniel Gonzalez se encargarán del rol de “Equipo de desarrollo”.

Una vez teniendo nuestros roles de trabajo nos enfocaremos a las fases que debemos de elaborar en el transcurso del desarrollo de una página web:

### 1. Sprint

Es el contenedor de los demás hitos del proceso. Todo lo que ocurre en una iteración para entregar valor está dentro de un sprint. La duración máxima es de un mes, el tiempo se determina en base al nivel de comunicación que el cliente quiere tener con el equipo. Los sprints largos pueden hacer que se pierda feedback valioso del cliente y poner en peligro el proyecto.

### 2. Sprint planning

En esta reunión todo el equipo Scrum define qué tareas se van a abordar y cuál será el objetivo del sprint. La primera reunión que se hace en el sprint puede llegar a tener una duración de 8 horas para sprints de un mes.

En la reunión debemos de hacer ciertas preguntas para recabar la información importante para la elaboración:

¿Qué se va a hacer en el sprint? En base a ello, se eligen tareas del Product backlog.

¿Cómo lo vamos a hacer? El equipo de desarrollo define las tareas necesarias para completar cada ítem elegido del Product Backlog.

En este punto nuestro equipo deber tener en claro un objetivo y se debe de encontrar comprometido con la entrega de valor que se hará al cliente al final del sprint. A esto se le llama sprint goal.

El resultado de esta reunión es el sprint goal y un sprint backlog.

### 3. Daily meeting

Es una reunión diaria dentro del sprint que tiene como máximo 15 minutos de duración. En ella debe participar, sí o sí, nuestro equipo de desarrollo y el Scrum Master. El Product Owner no tienen necesidad de estar presente.

En esta reunión diaria nosotros nos debemos de hacer algunas preguntas, las cuales son.

¿Qué hice ayer?

¿Qué voy a hacer hoy?

¿Tengo algún impedimento que necesito que me solucionen?

Esta reunión es la más oportuna para poder inspeccionar el trabajo y poder adaptarse en caso de que haya cambio de tareas dentro de un sprint.

#### **4. Sprint review**

La review del valor que vamos a entregar al cliente se hace en esta reunión, al final de cada sprint. Su duración es de 4 horas para sprints de un mes, y es la única reunión de Scrum a la que puede asistir el cliente. En ella el Product Owner presenta lo desarrollado al cliente y el equipo de desarrollo muestra su funcionamiento. El cliente valida los cambios realizados y además brinda feedback sobre nuevas tareas que el Product Owner tendrá que agregar al Product backlog.

#### **5. Sprint retrospective**

La retrospectiva es el último evento de Scrum, tiene una duración de 3 horas para Sprints de un mes, y es la reunión del equipo en la que se hace una evaluación de cómo se ha implementado la metodología Scrum en el último sprint.

Es una gran oportunidad para nuestro equipo Scrum de inspeccionar a sí mismo, proponiendo mejoras para el siguiente sprint.

**El resultado:** Una lista de mejoras que debe aplicar el siguiente día, ya que, al finalizar la retrospectiva, inmediatamente comienza un nuevo sprint, que incluye el sprint planning, daily meeting, sprint review y el ya mencionado sprint retrospective.

Para estas 5 partes podemos ayudarnos utilizando herramientas que nos faciliten el trabajo las cuales son 2 herramientas funcionales:

#### **1. Product backlog**

Product Backlog y sprint backlog: Metodología Scrum

Básicamente, el product backlog es el listado de tareas que engloba todo un proyecto. Cualquier cosa que debemos hacer debe estar en el product backlog y con un tiempo estimado por el equipo de desarrollo.

La responsabilidad exclusiva de ordenar el product backlog es del Product Owner, que se encuentra en constante comunicación con el cliente para asegurarse de que las prioridades están bien establecidas.

La ordenación también es 100% responsabilidad del Product Owner, por lo que las tareas que están más arriba deben de ser las de mayor prioridad.

El equipo de desarrollo elige tareas del product backlog en el sprint planning para generar tanto el sprint backlog como el sprint goal.

## **2. Sprint backlog**

Es el grupo de tareas del product backlog que el equipo de desarrollo elige en el sprint planning junto con el plan para poder desarrollarlas. Debe ser conocido por todo el equipo, para asegurarse de que el foco debe estar en este grupo de tareas.

El sprint planning no cambia durante el sprint, solo se permite cambiar el plan para poder desarrollarlas.



## JUSTIFICACIÓN Y DIAGRAMA DE ARQUITECTURA BASADA EN EL MODELO MVC

**Justificación:** El modelo vista controlador o conocido como MVC es un patrón de diseño arquitectónico de *software*, que sirve para clasificar la información, la lógica del sistema y la interfaz que se le presenta al usuario. En este tipo de arquitectura existe un sistema central o controlador que gestiona las entradas y la salida del sistema, uno o varios modelos que se encargan de buscar los datos e información necesaria y una interfaz que muestra los resultados al usuario final. Es muy usado en el desarrollo web porque al tener que interactuar varios lenguajes para crear un sitio es muy fácil generar confusión entre cada componente si estos no son separados de la forma adecuada. Este patrón permite modificar cada uno de sus componentes si necesidad de afectar a los demás. Por estos motivos utilizamos este modelo ya que es muy eficaz a la hora de utilizarlo en un proyecto y nos ahorra suficiente tiempo si es necesario modificar los componentes más fáciles.

### Modelos

Es la capa donde se trabaja con los datos, por tanto, contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas y harán los correspondientes *selects*, *updates*, *inserts*, etc.

### Vistas

Las vistas, como su nombre nos hacen entender, contienen el código de nuestra aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que nos permitirá renderizar los estados de nuestra aplicación en HTML. En las vistas nada más tenemos los códigos HTML y PHP que nos permite mostrar la salida.

En la vista generalmente trabajamos con los datos, sin embargo, no se realiza un acceso directo a éstos. Las vistas requerirán los datos a los modelos y ellas se generará la salida, tal como nuestra aplicación requiera.

### Controladores

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información, etc.

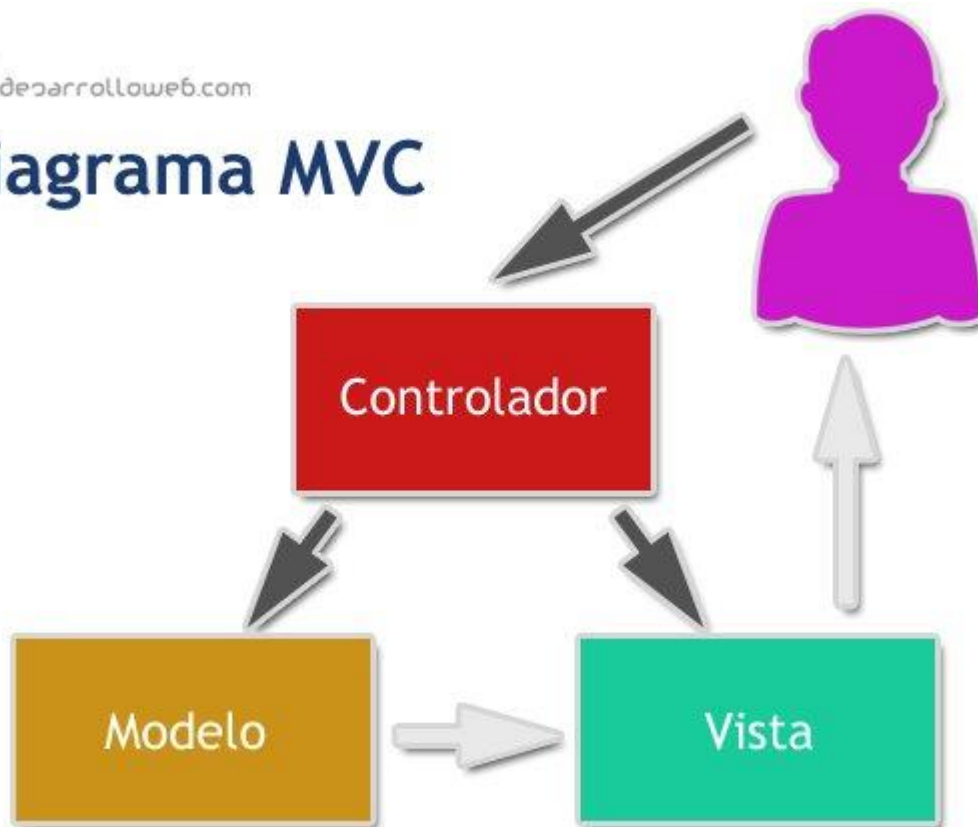
En realidad, es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo.

## Arquitectura de aplicaciones MVC

Diagrama que contiene la interacción de los componentes en el modelo MVC



### Diagrama MVC



En esta imagen hemos representado con flechas los modos de colaboración entre los distintos elementos que formarían una aplicación MVC, junto con el usuario. Como se puede ver, los controladores, con su lógica de negocio, hacen de puente entre los modelos y las vistas. Pero además en algunos casos los modelos pueden enviar datos a las vistas.

Mostraremos un ejemplo de como funcionaria el modelo MVC en uno de nuestros proyectos.

1. El usuario **realiza una solicitud** a nuestro sitio web. Generalmente estará desencadenada por acceder a una página de nuestro sitio. Esa solicitud le llega al controlador.
2. El **controlador comunica tanto con modelos como con vistas**. A los modelos les solicita datos o les manda realizar actualizaciones de los datos. A las vistas les solicita la salida correspondiente, una vez se hayan realizado las operaciones pertinentes según la lógica del negocio.
3. Para producir la salida, en ocasiones las **vistas pueden solicitar más información a los modelos**. En ocasiones, el controlador será el responsable de solicitar todos los datos a los modelos y de enviarlos a las

vistas, haciendo de puente entre unos y otros. Sería corriente tanto una cosa como la otra, todo depende de nuestra implementación; por eso esa flecha la hemos coloreado de otro color.

4. **Las vistas envían al usuario la salida.** Aunque en ocasiones esa salida puede ir de vuelta al controlador y sería éste el que hace el envío al cliente, por eso he puesto la flecha en otro color.

## JUSTIFICACIÓN DEL ENTORNO DE DESARROLLO WEB A UTILIZAR, ADEMÁS DE INCLUIR LA ACTIVIDAD DE CONFIGURACIÓN



**PHP** es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario.

Entre los factores que hicieron que PHP se volviera tan popular, se destaca el hecho de que es de código abierto.

Esto significa que cualquiera puede hacer cambios en su estructura. En la práctica, esto representa dos cosas importantes:

es de código abierto, no hay restricciones de uso vinculadas a los derechos. El usuario puede usar PHP para programar en cualquier proyecto y comercializarlo sin problemas.

está en constante perfeccionamiento, gracias a una comunidad de desarrolladores proactiva y comprometida.

### **Justificación:**

La programación de la página está hecha del lenguaje de php, como equipo optamos por elegir este lenguaje ya que es uno de los más usados, además de que es uno de los más fáciles de comprender.

**MySQL** es un sistema de gestión de bases de datos que cuenta con una doble licencia. Por una parte, es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle. Actualmente, es la base de datos de código abierto más famosa y utilizada en el mundo entero.



### **Justificación:**

En la base de datos se optó por utilizar este gesto de BD ya que aparte de contar con doble licencia es la más utilizada y mejor del mundo, además de su fácil comprensión en el apartado de crear, modificar y eliminar datos, además de que tiene una muy buena seguridad.



**JavaScript** es el lenguaje de programación encargado de dotar de mayor interactividad y dinamismo a las páginas web. Cuando

JavaScript se ejecuta en el navegador, no necesita de un compilador. El navegador lee directamente el código, sin necesidad de terceros. Por tanto, se le reconoce como uno de los tres lenguajes nativos de la web junto a HTML (contenido y su estructura) y a CSS (diseño del contenido y su estructura).

**Justificación:** java script fue implementado, ya que Con este lenguaje de programación del lado del cliente (no en el servidor) podemos crear efectos y animaciones sin ninguna interacción, o respondiendo a eventos causados por el propio usuario tales como botones pulsados y modificaciones.

**HTML** significa lenguaje de marcado de hipertexto, y le permite al usuario crear y estructurar secciones, párrafos, encabezados, enlaces y elementos de cita en bloque (blockquotes) para páginas web y aplicaciones.

HTML no es un lenguaje de programación, lo que significa que no tiene la capacidad de crear una funcionalidad dinámica. En cambio, hace posible organizar y formatear documentos, de manera similar a Microsoft Word.

En general, HTML es un lenguaje de marcado que es realmente sencillo y fácil de aprender incluso para los principiantes absolutos en la construcción de sitios web.

**Justificación:** HTML es esencial la creación de un documento HTML. Dicho documento se trata de un archivo de texto que cuenta con la extensión .html o .htm, donde se escribe todo el texto, así como las etiquetas HTML necesarias para la elaboración de una página web.



**CSS** es un lenguaje de hojas de estilo en cascada y se usa para estilizar elementos escritos en un lenguaje de marcado como HTML. CSS separa el contenido de la representación visual del sitio.

El código CSS hace la vida más fácil al desarrollador front-end al separar la estructura de un documento HTML de su presentación. Dicho de otro modo, el HTML actuaría como es esqueleto de la web, definiendo su estructura básica, y el CSS añadiría toda la capa de personalización sobre el que la web define su aspecto final.

**Justificación:** Se utilizó la tecnología de CSS para crear hojas de estilo, ya que cuando se hacen y se implementan, se hace notar una mejora en el aparatado de apariencia de nuestra página web.