**INSTITUTO SUPERIOR TECNOLÓGICO DEL SUR**

**CARRERA**

**DISEÑO Y PROGRAMACIÓN WEB**

**LENGUAJE WEB 2**

**“Web Frameworks”**

**PROFESOR(A) : Amado Cerpa Juan Andrés**

**ALUMNO : Vilca Apaza Christian**

**SEMESTRE : V**

13/11/2020

**Contenido**

[**Concepto** 3](#_Toc56112174)

[**Diferentes Frameworks** 4](#_Toc56112175)

[Django 4](#_Toc56112176)

[Flask 4](#_Toc56112177)

[Pyramid 5](#_Toc56112178)

[Web2py 5](#_Toc56112179)

[**Usos y Aplicaciones** 6](#_Toc56112180)

[**Principales Características** 6](#_Toc56112181)

[**Ventajas de los frameworks web** 7](#_Toc56112182)

[**Desventajas de los frameworks web** 8](#_Toc56112183)

[**Beneficios de los frameworks web** 9](#_Toc56112184)

[Rapidez 9](#_Toc56112185)

[Orden 9](#_Toc56112186)

[Soporte 9](#_Toc56112187)

[**Recomendaciones** 9](#_Toc56112188)

[**Conclusiones** 10](#_Toc56112189)

[**Bibliografía** 11](#_Toc56112190)

# **Concepto**

En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Los frameworks de lado servidor (es decir, "los frameworks de aplicaciones web") son frameworks software que hacen más fácil escribir, mantener y escalar aplicaciones web. Proporcionan herramientas y bibliotecas que simplifican tareas comunes de desarrollo web, incluyendo enrutado de URLs a los manejadores apropiados, interactuación con bases de datos, soporte de sesiones y autorizaciones de usuario, formateado de la salida (ej, HTML, JSON, XML), y mejora de la seguridad contra los ataques web.

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema.

# **Diferentes Frameworks**

Existen varios tipos de frameworks Web: orientados a la interfaz de usuario, como Java Server Faces, orientados a aplicaciones de publicación de documentos, como Coocon, orientados a la parte de control de eventos, como Struts y algunos que incluyen varios elementos como Tapestry.

La mayoría de frameworks Web se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC o con el modelo 2 de Servlets y JSP, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación.

En el caso de Python tenemos framworks en una gran variedad:

## Django

Django es un framework de aplicaciones web gratuito y de código abierto (open source) escrito en Python. Un framework web es un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente.

Se apoya en una comunidad grande y activa. Cuenta con una potente interfaz de administración, así como otras muchas características que deberás ir descubriendo tú mismo. Django ofrece formularios model-based, tiene su propio lenguaje de plantillas y cuenta con una excelente documentación que está disponible de manera gratuita. Técnicamente, Django utiliza un ORM (Object Relational Mapper) para asignar sus objetos a tablas de base de datos. El mismo código funciona con diferentes bases de datos y facilita la portabilidad entre diferentes tipos de bases de datos. Django trabaja con PostgreSQL, MySQL, SQLite y Oracle. Otras bases de datos se pueden utilizar con controladores de terceros.

Django no es un sistema de gestión de contenidos (CMS) como Joomla, Drupal o WordPress; es necesario escribir el código para hacer una web que trabaje. Django se utiliza en sitios de alto tráfico como Disqus, Pinterest, Instagram y Mozilla.

## Flask

Flask es un microframework que se creó originalmente como una broma del April Fools Day (como el día de los inocentes en EEUU) que derivó en un framework en solo un único archivo. Su principal intención es ser simple y pequeño; todo el framework consiste en un grupo de módulos. No hay un esqueleto o una estructura de la cual partir, todo se empieza con una página en blanco. Flask no proporciona grandes funcionalidades, pero hay extensiones Flask disponibles para agregar ORM, validación de formularios, manejo de carga, etc.

Flask es ideal, entre otras cosas, para aprender a programar y para ser utilizado por desarrolladores que se preocupan por las buenas prácticas y el código “elegante”, los que quieran crear prototipos de forma rápida y aquellos que necesitan una aplicación independiente. Flask se puede combinar con Jinja2, Mako, SQLAlchemy, Peewee, CouchDB, etc. para potenciar su funcionamiento.

## Pyramid

Pyramid nació de la fusión entre Pylons 1.0 y repoze.bfg. Este framework viene con “pilas incluidas”, pero no hace ninguna suposición acerca de los componentes de tu sitio web. La comunidad Pyramid está creciendo rápidamente y cada día son más los desarrolladores que se suman al uso del framework. La documentación es excelente y permite a los desarrolladores avanzar sin tener que contar con el apoyo de la comunidad. Pyramid se esfuerza por ser minimalista, rápido y fiable. Fue uno de los primeros frameworks web que fue compatible con Python 3. Se considera la mejor opción si deseas un inicio rápido, si trabajas en proyectos con la API, para prototipar un concepto o para el desarrollo de aplicaciones web grandes, como un CMS o un KMS.

## Web2py

El desarrollador líder de Web2py es Massimo Di Pierro, un profesor de Ciencias de la Computación que tiene su sede en Chicago. La plataforma tiene una documentación excelente, incluyendo un PDF de 600 páginas, con una introducción a Python. Si estás empezando en el desarrollo de Python, tendrás muchas preguntas, y Web2py puede tener respuestas para casi todo. Incorpora formularios generados que tienen validación de campo; las sesiones se almacenan del lado del servidor. El soporte de bases de datos incluye generación de SQL incorporada para las diez bases de datos más populares y Google App Engine.

Arquitectónicamente, Web2py sigue el Modelo-Vista-Controlador (MVC). No sigue la ruta de PHP de etiquetas incrustadas, sino que en su lugar genera HTML a partir del código. Una vez que haya definido sus modelos, obtendrá una interfaz administrativa completa de forma gratuita.

Y muy aparte tenemos mas frameworks web como CubicWeb, Dash, Glotto, Grok, Pylons, Reahl, Websauna, Wheezy web, Zope, Zope3, y la lista sigue y sigue ya que Python es muy popular.

# **Usos y Aplicaciones**

En resumida cuenta el uso de un framworkS nos ayuda a agilizar, organizar y administrar el desarrollo de un sistema web, ya que esta cuenta ya con todas las herramientas para empezar a desarrollar el sistema web, como dice el dicho no reinventes la rueda, si algo ya este hecho es mejor usarlo y mejorarlo.

Los frameworks se usa prácticamente en todo sistema web, por que ofrece seguridad y continúa mejorándose, los frameworks web se pueden aplicar también al diseño y desarrollo de paginas web modernas y mas rápidas, pasamos de usar html a usar un conjunto de herramientas que aparte de agilizar el proceso mejora la seguridad y la organización del desarrollo de páginas web.

# **Principales Características**

Casi todos los frameworks comparten las mismas características de acuerdo a su tipo, entre las que podemos destacar están:

La Autenticación mediante login y password que permite restringir el acceso y el tipo de permiso.

El Acceso a los datos en archivos txt, xml por ejemplo mediante interfaces que integran la base de datos.

Abstracción de URLs y Sesiones ya que el framework se encarga de manejarlas.

Internacionalización que permite la inclusión de varios idiomas en el desarrollo.

Controladores fácilmente adaptables a las necesidades del proyecto que gestionan las peticiones y/o eventos.

Inclusión del modelo MVC o versiones mejoradas de este.

Seguridad en cuanto a protección de datos.

Control de errores muy bien implementados.

# **Ventajas de los frameworks web**

Estructura y organización del código predeterminada. Los frameworks proporcionan tanto un esqueleto como una forma de trabajar. Por lo tanto, evitan tener que realizar un análisis sobre dónde situar los diferentes archivos de la aplicación (recursos, controladores, vistas, modelos, etc.).

Reutilización del código. Evitar duplicidad de código. En el desarrollo de una aplicación existen ciertos apartados que suelen repetirse, como la conexión con la base de datos, validación de formularios, páginas de estilos, etc. Con la utilización de un framework ahorraremos tiempo en desarrollar funcionalidades que ya están cubiertas y podremos enfocarnos en el funcionamiento de la aplicación más que en cómo llevarla a cabo.

Agilidad y rapidez en el desarrollo. Precisamente gracias a la reutilización de código mencionada anteriormente, conseguimos mayor rapidez en el desarrollo, ya que no perderemos tiempo en desarrollar funcionalidades nuevas.

Menor coste en el desarrollo. El coste es un parámetro que está directamente relacionado con la rapidez y agilidad. Acabar antes un proyecto implica que la dedicación es menor y por lo tanto el coste del proyecto también disminuye. Al igual que la rapidez en el desarrollo, esta ventaja beneficia tanto al cliente como al desarrollador.

Buenas prácticas de desarrollo con el uso de patrones. La mayoría de frameworks están basados en patrones de diseños, que nos indican pautas sobre cómo solucionar un problema específico que ya ha ocurrido con anterioridad. El patrón de diseño más popular es MVC (Modelo-Vista-Controlador), que nos ayuda a separar la capa de datos de la lógica del negocio de la interfaz con el usuario.

Minimizar errores y mayor facilidad para solucionarlos. Como el framework ya incorpora código implementado por otros programadores, los posibles errores que este pueda tener siempre serán menores que al desarrollarlo desde cero. Además, en caso de que hubiera un error, lo más probable es que ya haya sido solucionado por la comunidad.

Facilidad a la hora de encontrar una librería o código que ya cubra funcionalidades de tu desarrollo. Resulta más fácil encontrar herramientas (utilidades, librerías) adaptadas al framework que para un desarrollo propio.

Facilita la colaboración con otros desarrolladores. Tanto si son compañeros de tu equipo como de GitHub, leer el código desarrollado por otra persona puede resultar complejo. Sin embargo, si ya sabes qué estructura va a seguir el código y cómo se organiza, resultará más fácil comprenderlo y poder aplicarle nuevos cambios. Lo que nos lleva a la siguiente ventaja.

Facilita el mantenimiento. Si todos los miembros de un equipo trabajan de la misma forma, en el momento que haya que actualizar la aplicación o realizar algún evolutivo, tardaremos menos tiempo y el coste será menor.

# **Desventajas de los frameworks web**

Tiempo de aprendizaje. Antes de empezar a utilizar un framework debemos familiarizarnos con él, con cómo se estructuran sus archivos, con la forma en la que se comunican los componentes, etc. Por lo tanto, tendremos que invertir tiempo en superar la curva de aprendizaje para poder comenzar un nuevo desarrollo utilizando el framework.

Versiones inestables. El hecho de que los frameworks sean tan populares provoca que estén en constante actualización para cumplir con las nuevas tecnologías y las nuevas políticas de seguridad. Por ello, si en el desarrollo surgen incompatibilidades con otras librerías o se detectan errores de seguridad, la elección de una versión muy reciente del framework podría ralentizarnos.

Menor rendimiento. Los frameworks consumen, en general, más recursos que una aplicación creada desde cero y orientada al rendimiento. En aplicaciones muy exigentes, un framework puede resultar poco apropiado.

Código sin utilizar. Si la aplicación es pequeña o no requiere mucha funcionalidad, probablemente estaremos desaprovechando mucho código que viene implementado en el framework, por lo que la aplicación ocupará más espacio del que realmente necesitaría.

Elección del framework. Puede resultar complicado elegir cuál es el framework en el que te vas a especializar, ya que existe una gran variedad, y cada uno posee sus propias características. Por lo que, antes de lanzarse a aprender un framework nuevo, hay que dedicar tiempo a estudiar cuáles son los que tienen una base más sólida, ya que puede ocurrir que tras comenzar a desarrollar con un framework poco usado, se quede desactualizado al poco tiempo.

# **Beneficios de los frameworks web**

## Rapidez

Dentro de un framework encontrarás módulos — herramientas listas para ser utilizadas en casos específicos como conexiones a base de datos, vistas, controladores, ruteo de peticiones HTTP, cache para objetos, decorators, componentes HTML-CSS entre otros. Estos módulos te permitirán dar pasos gigantes a la hora del desarrollo, cubrirán los escenarios típicos, y lo mejor de todo, es que las herramientas están pensadas para mantener el orden en tu código.

## Orden

Si las necesidades de tu proyecto están cubiertas por las herramientas del Framework utilizado, este punto se cumple por la naturaleza del mismo. De lo contrario, como desarrollador, te encontrarás en la situación de proponer e implementar parte de la arquitectura del sistema. Cuando esto sucede, lo más recomendable es seguir las convenciones del Framework en cuestión, nombre de carpetas, archivos y separación de configuraciones.

## Soporte

La comunidad de desarrolladores son el verdadero soporte de los Frameworks de hoy en día. El impulso que se obtiene a través de sitios como [GitHub](https://github.com/) es una mejora continua

# **Recomendaciones**

Una vez se tenga encuenta que es un framework y para qué sirve existen recomendaciones antes de utilizar uno:

Para utilizar un framework es necesario conocer el lenguaje de programación en el que esta escrito este, así puedes entender muy rápidamente como funciona y para que circunstancias serviría usarlo.

Siempre se debe leer la documentación del framework a usar para asi tener una idea y una ayuda general para poder usar el framework mejor.

Escoger un framework es difícil, pero si te basas en que es lo que necesitas para tu sistema web podrás encontrar uno a tu medida.

Cada lenguaje de programación tiene un framework desarrollado para todo mundo y es el que a veces va en la cabeza en los listados, si hay alguna duda es mejor empezar a aprender el que mas se utiliza de cada lenguaje de programación.

# **Conclusiones**

Para concluir, un framework web es de gran utilidad en cuanto a rapidez, organización, seguridad y soporte.

Tiene grandes ventajas en cuanto al desarrollo de un sistema web, tiene gran variedad para escoger, cada framework tiene una curva de aprendizaje diferente hay algunos más fáciles que otros de aprender, hay algunos mas complejos robustos y mucho mas seguros que otros pero al final todos nos sirven para agilizar y ayuda a enfocarnos a solo programar lo que necesitamos para nuestros sistemas web.

# **Bibliografía**

<https://blog.guebs.com/2013/11/11/consejos-elegir-framework-php/>

<https://medium.com/@emilioborraz/beneficios-de-utilizar-un-framework-79753baac0ff>

<https://www.tithink.com/es/2018/08/29/framework-o-librerias-ventajas-y-desventajas/>

<http://snte-tics-educacion.blogspot.com/2016/09/framework-que-es-caracteristicas-y.html>

<https://es.ryte.com/wiki/Framework>

<https://openwebinars.net/blog/los-4-mejores-frameworks-para-aplicaciones-de-python/>

<https://wiki.python.org/moin/WebFrameworks>

<https://steelkiwi.com/blog/top-10-python-web-frameworks-to-learn/>

<https://es.wikipedia.org/wiki/Flask>

<http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf>

<https://es.wikipedia.org/wiki/Framework_para_aplicaciones_web>

<https://developer.mozilla.org/es/docs/Learn/Server-side/Primeros_pasos/Web_frameworks>