**¿Cómo crear un software? ¡Desde videojuegos hasta apps de celular!**



**Hugo Rodríguez - 30 Ene 22**

Articulo

7 min.

**Índice**

Aprender **cómo crear un software**quizá pueda parecer un proceso bastante complejo, dado que se requiere de conocimiento técnico general pero te aseguramos que, con los conocimientos necesarios y la práctica constante, no deberás tener inconveniente para diseñar proyectos exitosos.

En este momento, quizás te estás preguntando:

* ¿Cómo crear un software contable?
* ¿Cómo crear un software educativo?
* ¿Cómo crear un software de [**base de datos en la nube**](https://www.crehana.com/blog/tech/base-datos-nube/)?

Si estás dispuesto a desarrollar productos relacionados con el [**hardware y el software**](https://www.crehana.com/blog/desarrollo-web/que-es-hardware-y-software/) que mejoren las experiencias de los clientes, entonces, **estarás aportando al mercado productos innovadores y ricos en usabilidad y rapidez para que las operaciones sean más eficientes, seguras y productivas**.

Como comenta Dibbe Edwards, vicepresidente de la empresa [**de IBM**](https://www.ibm.com/blogs/internet-of-things/connected-products-solutions/) sobre cómo crear un software, “*el software ha surgido como un diferenciador clave en muchos productos, desde automóviles hasta lavadoras, con un creciente internet de las cosas que los conecta*”.

En este post, te contaremos cómo crear un software desde cero, cuáles son las fases del proceso de desarrollo de un proyecto de software y qué modelo elegir para tu proyecto. ¡Acompáñanos!​

**1. Define el tipo de software que quieres desarrollar**

Cuando hablamos de crear un software, podemos referirnos a una alta variedad de productos informáticos, desde videojuegos hasta apps y programas importantes del sistema. Por ese motivo, es importante definir cuál será la clase de programa que pretendes desarrollar.

Hay dos categorías bajo las cuales puedes establecer qué clase de porgramación pretendes crear:

* **Sistema de aplicaciones**: se centra en la creación de programas que satisfagan las necesidades de los usuarios.
* **Desarrollo del sistema**: se especializa en la creación y el mantenimiento de sistemas operativos mediante el desarrollo del ciclo de vida.

Los [**tipos de software**](https://www.crehana.com/blog/desarrollo-web/tipos-de-software/) más populares en la actualidad son los siguientes:

**Software de sistemas**

Los software de sistemas, también conocidos como operativos, son todos aquellos recursos y programas principales de un sistema operativo. **La ejecución de este tipo de funcionalidades se realiza por encima del resto del software, aunque depende del sistema operativo**.

Entre estos sistemas operativos, están los siguientes:

* Windows
* MacOS
* Ubuntu
* Unix
* Solaris
* Fedora

Dentro de este tipo de software de sistemas, también pueden estar incluidos los controladores de dispositivo o las herramientas de diagnóstico, por ejemplo.

**Software de programación**

En cuanto a los softwares de programación, suelen ser los más comunes para los programadores y profesionales en el campo de la informática. Básicamente, **abarcan todas aquellas herramientas con las cuales los profesionales buscan crear un nuevo software**.

¿Qué clase de herramientas? Por ejemplo:

* Editores de texto
* Compiladores
* Enlazadores
* Intérpretes
* Depuradores

**Aprende sin limites y alcanza tu mejor versión**

Prueba gratis

**Software de aplicación**

**Los softwares de aplicación son todos aquellos programas que se utilizan para habilitar cualquier tipo de tarea dentro de una computadora**. Por esta razón, son los tipos de software más populares.

Aquí encontrarás algunas plataformas clásicas como lo son:

* Los procesadores de texto.
* Los editores (por ejemplo, Photoshop).
* Las hojas de cálculo.
* Los programas de diseño.
* Los programas de comunicaciones.

**Software de desarrollo**

Aquí ya no podemos hablar acerca de cómo crear un software, pues**no se trata del tipo de programas que no se fabrican, sino que se desarrollan**. Aunque no es lo mismo que el desarrollo provenga de un programador o de un equipo de programadores (en cuanto a carga de trabajo), el procedimiento es similar.

Para elaborar un software de desarrollo en equipo, es muy común que uno de los desarrolladores sea quien se ocupe de la visión general del proyecto en cuestión, así como de las otras funciones y tareas de cada uno de los componentes.

Eso sí, independientemente de cualquier caso, el equipo debe mantenerse en comunicación constante para que el software final resulte funcional.

**Software libre**

No podíamos hablar acerca de cómo crear un software y dejar de mencionar al software libre. Este movimiento se desarrolló en 1985 y su desarrollo se ha convertido en sinónimo de libertad e independencia para muchas personas.

¿Por qué decimos esto? **Porque este software está ideado para que cualquier persona pueda acceder a él**. Tiene la opción de instalarlo, utilizarlo... ¡Y hasta transformarlo si así lo desean!



Fuente: Pexels

**2. Piensa en qué lenguaje de programación debes usar**

Es importante que, si estás considerando desarrollar tu mismo el programa, aprendas un lenguaje de programación. Cualquier persona puede tener ideas, pero **solo un desarrollador podrá convertir la idea en algo tangible**.

Específicamente, dentro de un lenguaje de software se encuentran los siguientes componentes:

* **Variable y vectores**: las variables vienen a ser una especie de contenedores de datos, mientras que los vectores son un conjunto de variables compuestas.
* **Condicionales**: aquellas premisas indispensables para que un programa pueda ejecutarse.
* **Bucles**: en caso de que exista una premisa, los bucles deben servir para la ejecución del código.
* **Funciones**: las variables que encierran un código dentro de sí mismas.

Incluso, si solo deseas trabajar en los aspectos del diseño del software, debes estar familiarizado con la codificación, como JavaScript o Python. Los lenguajes populares también incluyen C, C++ y C# para crear juegos, aplicaciones, controladores, sistemas operativos y otros ejemplos de proyectos de software para una empresa.

Recuerda que la etapa de escritura de código es, generalmente, la más larga, ya que es la columna vertebral de toda la fase del desarrollo del proyecto. Así que debes comenzar determinando el tipo de lenguaje que usarás para elaborar tu producto.

**3. Realiza una lluvia de ideas**

¿Quieres saber los parámetros que debes seguir para comenzar tu proyecto de software con buen pie? Te los contamos a continuación:

1. Primero, **es fundamental realizar un análisis empresarial**. Los analistas profesionales definen con precisión las necesidades del negocio y pueden recomendar una solución que aportará valor a todas las partes interesadas de la empresa sobre el desarrollo óptimo del software.
2. Luego, un segundo punto importante, es realizar un conjunto de tareas que permitan **crear una estrategia para el desarrollo de sistemas** que facilite y mejore los procesos, incorpore cambios significativos y cree nuevas políticas.
3. Un tercer punto que debes incluir en las actividades iniciales de la creación y desarrollo de un software, es **contar con un plan de trabajo** para el programa que deseas desarrollar y, además, debes incluir el propósito del programa y las funciones que quieres incluir.

Para llegar hasta este punto, te puede servir de gran ayuda realizar un *brainstorming*. **La creación de ideas innovadoras es un desafío, porque cualquier proyecto en mente debe ser cuidadosamente pensado para poder ser implementado con éxito**.

La planificación de proyectos de software es clave en esta etapa ya que genera los principales requisitos, el alcance del proyecto, define los medios, la disponibilidad de recursos y los problemas relacionados con los costos y plazos.

Si tienes dudas sobre cómo realizar una lluvia de ideas adecuadamente, te ofrecemos las mejores  [**herramientas de brainstorming,**](https://www.crehana.com/blog/empresas/herramientas-brainstorming-en-remoto/) para poder iniciar tus proyectos con buen pie.



Fuente: Pexels

**4. Asegúrate que tu programa cumpla con las características de un software**

Al momento de buscar crear un software, es inevitable preguntarnos qué características debe tener un software para ser considerado como tal.

Las características de los software se dividen en 3:

**Características operativas**

Estos rasgos refieren más que nada a la funcionalidad del software. Es decir, vendrían a ser la parte exterior del programa en cuestión.

Aunque no todos estos rasgos sean tomados en cuenta al momento de crear un software, existen ciertos aspectos que unifica este tipo de recursos, como por ejemplo:

* **Corrección**: debe cumplir los objetivos para los cuales fue creado.
* **Usabilidad**: fácil de aprender y entender.
* **Integridad**: es decir, que no sea un software con efectos secundarios.
* **Fiabilidad**: es decir, que el software no posea ningún defecto. Lo que vendría a ser el hecho de que no falle en la ejecución.
* **Eficiencia**: la manera en la cual el software utiliza los recursos que tiene a su disposición, como vendría a ser el caso del espacio de almacenamiento, por ejemplo.
* **Seguridad**: resistente a acciones y ataques externos.

**Características de transición**

Aquí englobamos aspectos destacados que pueden referir tanto a aspectos internos como externos de un software determinado. En ella se engloban elementos como los siguientes:

* **Interoperabilidad**: representa la disposición para intercambiar información con otras aplicaciones.
* **Reutilización**: poder usar el código de software, con diversas modificaciones y para propósitos diversos.
* **Portabilidad**: que resulte accesible desde diversos equipos.

**Características de revisión**

En este punto, englobamos aquellos elementos de ingeniería, lo que podría ser la calidad interior del software.Algunos elementos que se aprecian aquí son la documentación, estructura y eficiencia:

* **Capacidad de mantenimiento**: al momento de crear el software, este debe ser fácil de mantener.
* **Flexibilidad**: que tenga la disponibilidad para ser modificado por los desarrolladores.
* **Extensibilidad**: es fácil de incrementar nuevas funciones.
* **Escalabilidad**: el software debe ser fácil de actualizar para más trabajo.
* **Modularidad**: tiene que estar compuesto por módulos y unidades independientes entre sí.
* **Capacidad de prueba**: debe ser fácil la prueba del software.

Estas son las formas en las que se pueden calificar las características de este tipo de plataformas, que debes tener presente antes de crear un software.

**5. Apóyate en recursos en línea**

Existen muchos recursos que pueden ayudarte a saber cómo hacer un buen desarrollo de softwares y, gracias a internet, puedes encontrar cursos online que te permitirán aprender rápidamente mientras trabajas tus proyectos con programas para crear un software.

Estos son algunos cursos virtuales para aprender cómo crear un software desde cero:

* [**Curso online de software debugging: depura tus aplicaciones web**](https://www.crehana.com/cursos-online-tech/software-debugging-depura-tus-aplicaciones-web/)
* [**Curso online de fundamentos de JavaScript**](https://www.crehana.com/cursos-online-front-end/fundamentos-de-javascript/?keyword=java&source=search&model_used=SEARCH_ENGINE_V2.1&product_name=Fundamentos%20de%20JavaScript&product_id=9137&keyword=java&item_type=course&position_selected=2)
* [**Curso online de gestión ágil de proyectos**](https://www.crehana.com/cursos-online-innovacion/gestion-agil-de-proyectos/)
* [**Curso online de prototipado desde cero con Figma**](https://www.crehana.com/cursos-online-ui-ux/prototipado-desde-cero-con-figma/)



Fuente: Pexels

**6. Redacta la documentación del proyecto**

Una vez tengas claro cómo crear un software, comienza con la escritura de la documentación de tu proyecto. **Este archivo debe describir las características y lo que buscas lograr con el proyecto**.

Empezar con la documentación del software es una buena forma de trabajar los pasos de manera ordenada para el desarrollo del sistema. Además, te ayudará a mantenerte enfocado y encaminado.

Eso sí, antes de invertir tiempo y dinero en el proyecto, es importante realizar un análisis de la viabilidad del proceso de desarrollo del software.

El estudio de viabilidad puede mostrar cómo rentabilizar la creación del software a largo plazo evaluando todos los factores (económicos y técnicos) que posiblemente afecten al desarrollo del proyecto. Y esto lo puedes estructurar al momento de realizar tu documentación respectiva.

Siempre regresa a la guía para obtener los detalles cuando escribas el proyecto.

**7. Determina el diseño que tendrá tu proyecto y elabora un prototipo**

El diseño es la fase del desarrollo de un proyecto que involucra la conceptualización del producto. **Se realiza de acuerdo a las especificaciones escritas durante las dos primeras etapas del desarrollo del software**.

Los diseñadores construyen la estructura completa del proyecto y proporcionan el prototipo final que se utilizará para los siguientes pasos del desarrollo del software.

Hacer un prototipo es un paso básico para crear un software que muestra las funcionalidades que pretendes conseguir al finalizar tu proyecto. Fundamentalmente, es un modelo que se desarrolla de manera rápida, y que debe repetirse hasta lograr que el software funcione correctamente.

Es normal que el prototipo cambie durante el ciclo del desarrollo de un programa a medida que surgen nuevas formas de abordar los problemas o nazcan nuevas ideas que desees incorporar al proyecto.



**Lista de etiquetas HTML y propiedades CSS**

Aprende cuáles son los términos básicos de los lenguajes de programación HTML y CSS. Descarga esta lista de etiquetas HTML y propiedades CSS gratis.

Descargar

**8. Haz pruebas de usabilidad**

Los errores son la pesadilla de todos los programadores, por eso, **siempre se recomienda hacer testing** (las veces que sean necesarias) dentro del desarrollo del software.

¿Temes que tu proyecto fracase o no comience teniendo el éxito que esperabas? La profesora Lorena Duarte, docente de nuestro  [**curso online de Prototipado y user testing con Design Thinking**](https://www.crehana.com/cursos-online-ui-ux/prototipado-y-user-testing-con-design-thinking/), sostiene que "*no es necesario ver una idea fracasar, basta con fallar en el momento adecuado*".

**9. Aplica la integración e implementación de software**

La integración de todas las fuentes y entornos es imprescindible para descubrir cómo crear un programa de software de manera eficiente.

**Este proceso ayuda a determinar a tiempo cuántos problemas, conflictos y errores existen en el proceso de desarrollo de software**. Ahora, la implementación es la fase responsable de la ingeniería del software sobre el control de calidad del código escrito por los desarrolladores.

En esta etapa del desarrollo del software, se utilizan diferentes marcos y tipos de prueba para saber si hay algún error en el sistema. La aplicación recién construida y probada se traslada a la producción, incluyendo la transferencia de datos y componentes.

**10. Lanza la primera versión**

Finalmente, se lanza la primera versión de la creación del software que incluye mantenimiento y actualizaciones periódicas.

Recuerda: **cada una de las etapas del ciclo de vida del software depende del modelo para el proceso de desarrollo de software**que elija cada empresa, como por ejemplo:

* El modelo cascada
* El modelo ágil

Como puedes ver, aprender cómo crear un software exitoso desde cero no representa gran dificultad. **Sólo necesitas constancia, esfuerzo y disciplina para aplicar los pasos correspondientes para diseñar un programa que esté a la altura de tu trabajo**.

¡Este ha sido nuestro post! Si quieres saber más acerca de cómo crear un software exitoso, inscríbete en nuestros [**cursos de desarrollo web y tecnología**](https://www.crehana.com/cursos-online-desarrollo-web-y-tecnologia/), donde aprenderás a desarrollar el mejor software del mercado y crear los mejores programas de la industria.

¡Éxitos!