

# Tengo que desarrollar un nuevo Software ¿por dónde empiezo?

Sin lugar a dudas, éste es el problema más frecuente al que nos enfrentamos los programadores: ¿por dónde comenzar a desarrollar una aplicación? Te enloqueces buscando consejos, tratando de elegir la metodología apropiada pero alguna vez ¿te preguntaste cuál es el método que mejor se ajusta a tus rasgos de personalidad? Este artículo abarca tanto aspectos técnicos y conceptuales como rasgos psicológicos relativos a cada individuo, que pueden ayudarte a tomar la decisión que mejor se ajuste a tu perfil.

Escrito por: **Eugenia Bahit** (Arquitecta GLAMP & Agile Coach)



Eugenia es **Arquitecta de Software**, docente instructora de tecnologías **GLAMP** (GNU/Linux, Apache, MySQL, Python y PHP) y **Agile coach** (UTN) especializada en Scrum y eXtreme Programming. Miembro de la **Free Software Foundation** e integrante del equipo de **Debian Hackers**.

**Webs:**

Cursos de programación a Distancia: [www.cursosdeprogramacionadistancia.com](http://www.cursosdeprogramacionadistancia.com)  
Web personal: [www.eugeniabahit.com](http://www.eugeniabahit.com)

**Redes sociales:**

Twitter / Identi.ca: [@eugeniabahit](https://twitter.com/eugeniabahit)

Cada vez que tenemos que comenzar un nuevo proyecto de desarrollo de Software, dos sentimientos encontrados se nos presentan: el entusiasmo por empezar un proyecto nuevo y la sensación de tener miles de ideas pero sin embargo, ni una que nos indique cómo iniciar el camino.

Son sensaciones mucho más comunes de lo que se cree y se presentan hasta en los desarrolladores más experimentados. El tema es ¿cómo abordar el problema? Si aún continúas experimentando estas sensaciones es hora de que comiences a prestarte más atención a ti mismo y te plantees qué camino se adapta mejor a tu personalidad para así, lograr alcanzar el éxito que tanto deseas y muchas veces te deja con esa sensación de no haberlo alcanzado.

Si estás por comenzar un nuevo proyecto, presta atención a las siguientes páginas y espera llegar al final del artículo para asegurarte tomar la decisión más acertada.

## 1. Describe la visión global de la aplicación

Lo primero que uno debe tener muy en claro es **“de qué se trata la aplicación que se necesita desarrollar”**. Tenerlo en claro, no significa describir con precisión -siquiera aproximada- las características de la aplicación ni mucho menos, sus funcionalidades. Por el contrario, implica tener una visión global de lo que se necesita.

Pero **¿qué significa entonces, tener una visión global de la aplicación?** Puedes imaginar esa “visión global” como una breve descripción que te ayudará a “vender” tu aplicación. Alguna vez habrás visto un programa de televisión, donde el conductor o conductora de TV te invita a probar determinado producto. Es sabido que el tiempo en televisión “es tirano” así que el conductor, “lee” un rápido anuncio. Ese rápido anuncio, a veces es solo “un gancho” para generarte curiosidad. Pero la mayoría de las veces, te deja muy en claro de qué se trata el producto. Y eso mismo, es la visión global que tu debes tener con respecto a la aplicación.

La visión global, entonces, debe ser una breve descripción que en lo posible, no supere un párrafo y a lo sumo, se concluya en dos. Incluso, más ideal es aún, poder describir la aplicación en una única frase.

Una forma simple de intentar obtener una visión global de la aplicación, sería imaginar el anuncio que harías para ofrecerla. Te doy algunos ejemplos:

1. Con AppName podrá llevar un control preciso en tiempo real, de los productos disponibles en su stock y obtener alertas inmediatas cuando el inventario de existencias sea bajo.
2. Ingresa a AppName.com y encuentra el curso que estás buscando. Docentes especializados en diversas áreas, te esperan para que asistas a sus clases en línea.
3. Encuentra amigos o citas en SiteName.com. Muchas personas esperan conocerte. Dinos cómo es la persona que deseas encontrar y te presentaremos a quienes se ajusten a tu perfil.

Diferentes plataformas, diferentes objetivos, diferente público. Sin embargo, no cuesta trabajo imaginarse de qué se trata cada una de las aplicaciones.

Una vez tienes “tu anuncio”, convertirlo en una descripción menos personal, no es difícil. Solo se trata de redactarlo de manera impersonalizada y tratando de reducir al máximo posible, los adjetivos que generan indirectamente, una opinión previa del producto:

1. Sistema de control de *stock* en tiempo real y alertas inmediatas de inventario.
2. Plataforma *online* de búsqueda y contratación de cursos a distancia sobre diversas áreas de conocimiento.
3. Plataforma de citas *online* con sistema inteligente de contraste de perfiles y

examen de compatibilidades.

A partir de tu visión global, te será mucho más simple entender qué es aquello que necesitas para alcanzar tus objetivos.

## 2. Describe las particularidades de tu aplicación, solo cuando te sientas seguro de poder hacerlo

Es muy importante poder entrar en cuestiones más particulares sobre la aplicación, antes de decidir cómo vas a desarrollarla y qué tecnologías implementarás.

Pero para entrar en cuestiones particulares, debes evitar los detalles y solo concentrarte en aquellas cosas que la aplicación podrá hacer, es decir, en la descripción general de sus funcionalidades.

Definir las generalidades de las funciones de un sistema, no es poca cosa. Es aquí donde los problemas de organización y análisis de requerimientos comienzan a salir a la luz.

Existen dos métodos probados que puedes emplear para esto, pero ninguno de ellos te asegura resultados óptimos. Por ello, **antes de decidir qué método deseas emplear, es muy importante que te tomes el tiempo de “definirte a ti mismo”**, ya que el resultado de cada método, **dependerá de tus fortalezas y debilidades**.

*Nunca olvides que conociendo tus debilidades, te abres camino para poder trabajarlas y convertirlas en fortalezas*

Los métodos que puedes emplear, son:

1. **Método tradicional:** consiste en hacer una lista de funcionalidades conocidas de la aplicación. Por ejemplo:
  - ABM de productos y categorías
  - Generación automática de reportes de *Stock*
  - Generación automática de alertas cuando el *stock* llegue a *N* productos

Este método es el más común, pero sin embargo, te enfrenta a ciertos desafíos difíciles de resolver a la hora de pasar a los detalles. Este método, **puede ser contraproducente en los siguientes casos:**

- Eres una persona que se distrae con facilidad;
- Generalmente, te cuesta organizarte;
- Suelen concentrarte mucho en los detalles y con frecuencia, pierdes el objetivo original de vista;
- Te sueles “enredar” a menudo cuando intentas planificar lo que debes hacer para cumplir un objetivo.

Por el contrario, el mismo método **puede ser productivo cuando:**

- Te resulta extremadamente imposible (o directamente imposible) convivir con incertidumbres (no saber o desconocer lo que sucederá);
- A menudo, sientes la necesidad de mantener el control sobre situaciones y/o personas que te rodean;
- Con frecuencia, puedes anticiparte a las consecuencias resultantes de tus decisiones;
- Frente a diferentes circunstancias sueles experimentar una especie de bloqueo frente a lo desconocido;
- Generalmente, tienes la sensación de no saber si eres o no capaz de lograr aquello que te propones.

2. **Método ágil:** consiste en definir las acciones que hará el sistema, desde la óptica de quiénes lo utilizarán. Para ello, se utiliza un formato descriptivo llamado “Historia de Usuario”. Una historia de usuario se arma con la frase: *como [mi rol de usuario] puedo [acción]*. Opcionalmente, puede completarse con: *para [beneficio]*. Por ejemplo:

- Como administrador del sistema puedo agregar categorías
- Como administrador del sistema puedo editar categorías
- Como administrador del sistema puedo eliminar categorías preexistentes
- Como responsable de depósito puedo ingresar un producto al inventario
- Como responsable de depósito puedo obtener un reporte del stock de los productos existentes en el depósito
- Como vendedor puedo consultar el *stock* de un determinado producto

Este método, es mucho menos utilizado que el tradicional. Sin embargo, estadísticamente ha demostrado arrojar mejores resultados en el software final generado. No obstante, **puede presentarse como un obstáculo, cuando:**

- A menudo y en diferentes situaciones experimentas sensaciones de

ansiedad sin origen aparente;

- Suelen experimentar con mucha frecuencia una sensación de bloqueo, estancamiento, avance lento o crisis de ansiedad frente a circunstancias no previstas;

A diferencia del anterior método, este otro puede resultar **sumamente productivo** una vez se supera (o se aprende a manejar) el miedo a lo desconocido, sin mayores requisitos. No obstante, ciertas características a tener en cuenta al momento de elegir esta opción, son:

- Eres una persona segura de sí misma y con frecuencia sientes que puedes alcanzar los objetivos que te propones;
- Te estresas con facilidad frente a las grandes lluvias de ideas o grandes cantidades de información y te sientes mucho más cómodo y distendido, haciendo las cosas “paso a paso”;
- Tienes facilidad para manejar las sensaciones que te generan los cambios de planes o te adaptas con facilidad a los cambios sin que ello implique no poder continuar con tu vida diaria de forma natural.

*Elegir aquel método que mejor se adapte a tu personalidad, evitando ser tu quien deba adaptarse al método, será la principal fuente de garantía sobre los resultados que finalmente obtengas.*

### 3. Prioriza y valora las funcionalidades del sistema

En principio, debes tener en cuenta que hayas optado por un método tradicional o uno ágil, las funcionalidades del sistema pueden ir cambiando y/o ampliándose con frecuencia. No obstante, suelen tenerse mayor cantidad de funcionalidades descritas cuando se trabaja de forma tradicional que de forma ágil. Sin embargo, un buen número de funcionalidades descritas para poder comenzar con el desarrollo, suele ser inferior o igual a 10. Si puedes “frenar” un poco en ese punto, te será mucho más fácil avanzar en el desarrollo. Es en este punto donde puedes comenzar a priorizar y valorar las funcionalidades.

#### Prioridad

La prioridad de una funcionalidad, es una cuestión de dependencia lógica. La mejor estrategia para darle prioridad a una funcionalidad sobre otra, es preguntarse ¿puedo contar con la funcionalidad “X” sin depender de la funcionalidad “Y”? Cuando no sabes qué responder a esa pregunta, puedes llegar un poco más a fondo preguntándote:

¿depende “X” de alguna otra funcionalidad?

- Cuando “X” dependa de “Y”, “Y” tendrá prioridad sobre “X”.
- Cuando “Y” dependa de “X”, “X” tendrá prioridad sobre “Y”.
- Cuando “X” no dependa de “Y” e “Y” no dependa de “X”, podrá tener mayor prioridad, aquella que mayor valor tenga para la aplicación.

## Valor

El valor que una funcionalidad tenga para la aplicación, a diferencia de la prioridad, no es una cuestión de dependencia, sino de relevancia para el negocio. *¿Qué funcionalidad generaría más éxito a mi aplicación?* Se puede decir que el valor que uno le otorgue a una funcionalidad, será relativo a las consecuencias que dicha funcionalidad generará para el negocio. Por ejemplo, en un sistema de control de inventario, se cree que generará un mayor éxito a la aplicación la posibilidad de generar reportes de *stock* que la de administrar categorías de productos.

## Prioridad y valor: ¿cómo otorgarlos?

Al momento de dar prioridad y valor a un pequeño número de funcionalidades, deberás tener en cuenta lo siguiente:

- La prioridad es única para cada funcionalidad y se aplica de forma incremental. A menor número, mayor prioridad. Es decir que la funcionalidad con prioridad 1 será prioritaria frente a la de prioridad 2.
- La valoración se mide en base a una escala simbólica que puede ir de 1 a 100 en incrementos de 10 o 5. A mayor número, mayor será la relevancia de esa funcionalidad para el sistema. Pueden coexistir múltiples funcionalidades con la misma relevancia.

## 4. Detalla la funcionalidad con mayor prioridad

Una vez hayas priorizado y valorado al menos un total de entre 5 y 10 funcionalidades, puedes tomar aquella a la cuál le hayas asignado la mayor prioridad y comenzar a definir sus **criterios de aceptación**.

Los criterios de aceptación de una funcionalidad, abarcan hasta el más mínimo detalle sobre ésta. Los mismos, van desde aquello que verá el usuario al ingresar a dicha funcionalidad hasta los pasos que deberá seguir para concretar el proceso y la forma en la que el sistema deberá responder y reaccionar frente a dichas acciones del usuario.

Por ejemplo:

Tomemos el caso de la funcionalidad que tradicionalmente la hemos definido como Generación automática de reportes de Stock y ágilmente dijimos que Como

responsable de depósito puedo obtener un reporte del stock de productos.

Los criterios de aceptación para esta funcionalidad podrían describirse detalladamente como los siguientes:

- Desde la pantalla principal del sistema, el usuario deberá pulsar un botón con la leyenda “Generar reporte de Stock”.
  - El sistema deberá mostrar en la siguiente pantalla, una lista con los nombres de los productos, el código de barras, el sector del depósito en el cual se ubica cada producto (letra y número de sector) y la cantidad de existencias.
    - Cuando la existencia sea inferior o igual a 50, se deberá mostrar el stock en rojo.
    - Cuando la existencia sea menor a 25, se deberá mostrar un botón a la derecha del stock, con la leyenda “Alertar al responsable de depósito”.
      - Al pulsar dicho botón, se enviará un e-mail a la cuenta del responsable de depósito (responsable@example.org), con asunto “Existencia insuficiente del producto {código de barras del producto} (stock: {stock})”.
      - El e-mail no podrá enviarse con una frecuencia inferior a 24 horas.
      - Cuando dicha existencia insuficiente haya sido reportada, el sistema deberá quedar al pendiente y transcurridas 72 horas desde el envío de la primera alerta, deberá constatar si se han repuesto existencias. De no haberse repuesto, deberá enviar el mismo e-mail de forma automática, al responsable de depósito y al Gerente de Producción (gerente@example.org).

## 5. Define las tareas necesarias para que cada funcionalidad sea desarrollada

Cuando los criterios de aceptación de al menos 1 funcionalidad se encuentren definidos y 5 o 10 funcionalidades hayan sido descritas, estarás en condiciones de evaluar qué tecnologías y/o herramientas serán convenientes para el desarrollo de la aplicación. Para ello deberás “colocar en la balanza”:

- La visión global de la aplicación;
- El conjunto de funcionalidades ya descritas;
- La base de referencia de los criterios de aceptación especificados;
- Los recursos de los cuáles dispones;
- Tu nivel de conocimientos;
- El tiempo que puedes invertir en investigar antes de tomar la decisión tecnológica apropiada; y...
- Factores ajenos a tu responsabilidad o alcance, que impacten de forma directa sobre la aplicación.

Haciendo un balance de los 7 puntos anteriores, podrás tomar la decisión tecnológica más apropiada y cuando lo hagas, estarás en condiciones de comenzar a definir las tareas para iniciar el proyecto y las requeridas por cada una de las funcionalidades.

Las **tareas iniciales del proyecto**, suelen basarse en todo aquello que debes hacer para preparar tu entorno de desarrollo y posiblemente, el núcleo o motor bajo el cuál funcionará la aplicación. Estas tareas, suelen hacerse al inicio del proyecto y luego, en eventuales ocasiones, requerirán algún tipo de adaptación pero con poca frecuencia.

Las **tareas específicas a cada funcionalidad** son tal vez, las que mayor “ruido” generan al momento de plantearse “¿qué debo hacer para lograr desarrollar esta funcionalidad?”.

Básicamente, deberás escribir una lista (tal cuál vayan apareciendo las ideas), de cada una de las cosas que necesitas hacer para que dicha funcionalidad pueda ser desarrollada. Estas tareas pueden ser de diseño gráfico, maquetación, programación, arquitectura, DBA, *testing*, QA hasta incluso, tareas administrativas.

Hacer una lista independiente de las tareas necesarias para desarrollar cada una de las funcionalidades de la aplicación, te permitirá:

- **Discriminar y agrupar tareas por tipo de tarea:** por ejemplo, las de diseño gráfico, las de programación, las de investigación, las de documentación, etc.;
- **Organizar el orden en el que vas a llevarlas a cabo:** de la lista de tareas y a simple vista, podrás ver aquellas que necesites emprender antes que otras. No será difícil notarlo: bastará con que te preguntes *¿puedo hacer la tarea X antes que la tarea Y?* Si una tarea no depende de otra, la puedes desarrollar en el momento que desees. Guíate por el método que utilizaste al momento de priorizar funcionalidades y aplícalo a las tareas.

## ¿Cómo seguir?

Al **trabajar de manera tradicional**, los pasos 2, 3, 4 y 5 suelen realizarse **de forma consecutiva hasta alcanzar la última funcionalidad** y el último detalle requerido por ésta. Los mismos, son plasmados en un documento denominado “Documento de Alcance” y no se comienza con el diseño y desarrollo de la aplicación, hasta que dicho documento se encuentra finalizado y aprobado por los principales interesados en el proyecto.

Cuando **se trabaja de forma ágil**, el paso 2 se realiza en forma paralela a los siguientes e incluso, mientras que se está desarrollando la aplicación, al igual que sucede con el paso 3. Los pasos 4 y 5, se realizan cíclicamente. Al comienzo del proyecto, se establece la duración de un ciclo de desarrollo (por ejemplo, cada 2 semanas se decide cumplir un ciclo). Dependiendo de este tiempo, se eligen una, dos (o más) historias de usuario, se efectúan los pasos 4 y 5 al comienzo del ciclo, se diseñan y desarrollan las historias de usuario y luego, se repite el ciclo desde el paso 4.