



Universidad  
Andrés Bello®  
Conectar • Innovar • Liderar

# DESARROLLO DE PORTAFOLIO DE UN PRODUCTO DIGITAL

## El portafolio de productos

Portafolio es un término que engloba gran cantidad de significados, sin embargo, existen dos que definen mejor esta herramienta:

1. Se puede determinar según el conjunto de proyectos acabados, de trabajos realizados, de estudios, de maquetas y de todo aquello llevado a cabo por un profesional.
2. A través de un grupo de profesionales que avale su experiencia y preparación.

Para entender lo que es un portafolio, debemos primero hablar sobre los proyectos y programas que son la base para la creación de un portafolio.

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único y un programa es un grupo de proyectos relacionados entre sí, gestionados de manera uniforme para obtener beneficios y control, que no se obtendrían si se gestionan en forma individual.

Finalmente el portafolio se refiere a un conjunto de proyectos o programas y otros trabajos que se agrupan en un solo lugar para facilitar la gestión y visualización de estos.

Hay diferentes estrategias que se pueden utilizar para crear un portafolio que potencie tu perfil profesional, es decir, que podemos utilizar el formato que mejor se adapte dependiendo del tipo de trabajo que desees postular. También existen diferentes herramientas digitales que permiten crear un portafolio, ya sean programas como la Suite de Adobe, aplicaciones como Visual Studio Code, que permiten trabajar el diseño en código HTML y CSS o plataformas online como Behance que exhiben de manera gratuita tu portafolio.

## ¿Qué es un portafolio de productos?

Un portafolio de productos se puede definir como un resumen de los trabajos y proyectos que ha realizado una persona y tiene como objetivo mostrar una variedad de productos/servicios, proyectos y programas, que ha realizado a lo largo de su carrera.

Cuenta la historia sobre las decisiones que se tomaron para hacer un determinado trabajo, evidenciando los procesos que se llevan a cabo y la aplicación de diferentes herramientas, para conseguir los objetivos propuestos de cada proyecto, es a través de este recurso, se puede gestionar la visualización de los proyectos en plataformas digitales cuyas interfaces permiten generar un portafolio llamativo, interactivo que puede ser hasta colaborativo.

## Importancia de contar con un portafolio

Hoy en día los perfiles en la industria TI se encuentran cada vez más demandados y el modelo de negocio de las empresas se encuentran en un cambio constante adaptando sus equipos de trabajo hacia las metodologías ágiles, con el fin de adaptarse a un sistema que cada vez se encuentra más inmerso en la web.

Para esto, dejamos atrás el modelo convencional de mostrar nuestras capacidades a través de un curriculum y evidenciamos a través de un portafolio el trabajo que hemos realizado y también cómo aplicamos diferentes herramientas, estrategias, análisis y demás.

Un portafolio es como una vitrina de lo que es el trabajo personal y se pueden incluir los trabajos colaborativos o en equipo declarando la participación de terceros dentro del proyecto. Es por esto que es de suma importancia contar con un portafolio a la hora de postular a un trabajo ya que permite ver y analizar mejor las capacidades de las personas en la ejecución de proyectos.

## ¿Qué busca un reclutador en el Portafolio?

Por ejemplo, si eres un profesional que desarrolla sitios web, y te encuentras en la búsqueda de un trabajo relacionado a tu perfil, lo más seguro es que al momento de postular te hagan la siguiente pregunta, ¿Qué sitios web ha desarrollado? y para esto, además de mostrar tu perfil profesional y educacional a través de un currículum, debes contar con un portafolio que avale y muestre la expertis del área y señalar los sitios web en los que ha trabajado mostrando fotos del sitio web finalizado, también pueden ser fotos del proceso, herramientas de maquetación web y todo lo relacionado al área.

Sin duda esto facilita el proceso de selección ya que es más eficiente para el reclutador ver directamente trabajos o proyectos, si la visión que tiene la



persona o reclutador es acorde a los requerimientos de la empresa pueden definir si eres la persona que buscan.

¿Qué tipos de portafolios existen?

Ten en cuenta que los portafolios digitales no deben construirse de la misma manera. Estos varían dependiendo de tu especialidad. A continuación te explicamos algunos tipos de portafolios y el contenido que puedes incluir en ellos.

- **Portafolio de Dibujo, Ilustración y Arte digital**

Este tipo de portafolios profesionales se caracterizan, principalmente, por contener imágenes. Ahora, si eres una persona que se desempeña en diferentes campos, es importante que crees un portafolio diferente para cada disciplina. Bien sea que te desenvuelvas en, el diseño de personajes, arte conceptual, Ilustración editorial o fotografía, tener a disposición varios portafolios profesionales es una buena manera de organizar tus proyectos y la persona que te evaluará tendrá un mejor criterio acerca de tu especialidad.

- **Portafolio de Redactor**

Si eres redactor/a, alguna vez te habrán pedido que enseñes lo que escribes. Un portafolio profesional que contenga tus artículos o textos organizados es una excelente manera de mostrar tu trabajo. Además, existen diferentes plataformas online que te ayudarán a guardar tus textos de una manera atractiva. Tal vez para los redactores el portafolio profesional no es tan visual como para otras áreas artísticas, sin embargo, es importante que demuestres tus capacidades con una exposición en línea de tus mejores trabajos.

- **Portafolio desarrollo web**

Si trabajas creando sitios web, aplicaciones u otras áreas relacionadas al desarrollo web, la mejor manera de presentar tus proyectos es a través de una plataforma llamada GitHub, que permite generar repositorios de los sitios web en los que hayas trabajado evidenciando las habilidades en el código y manejo de el.

**Buenas prácticas para la creación de un portafolio**

Para gestionar un portafolio es necesario contar con una estructura apropiada. La buena organización es clave para gestionar el portafolio dentro de un marco claro.

Algunas de las actividades que se realizan para mantener una buena estructura son:

- Establecer cómo se van a seleccionar y dar prioridad a los proyectos, es decir, seleccionar los proyectos en los que hemos trabajado que tengan mayor relación al trabajo que se postula.
- Manejar las comunicaciones y rendimiento de los proyectos, utilizando un estilo gráfico unificado y coherente con el contenido, utilizar una paleta de colores acotada y presentar desde el minimalismo y simpleza los proyectos en cuestión, señalando el alcance que tuvo y resultados de los KPI.
- Se debe definir los procesos, es importante establecer ciertos hitos que enmarcan cada etapa del proyecto, estos pueden ser organizados cronológicamente a través de una línea del tiempo o según el criterio que englobe mejor cada área.

Para que sea efectivo y pueda cumplir con el objetivo, un portafolio debe incluir la descripción del rol que tiene la persona dentro de un proyecto, cómo se relaciona con los otros roles, cuales son los aprendizajes logrados desarrollando el caso, además de ser flexible y adaptarse a la oferta del trabajo. Se debe entender que busca la empresa que los recluta y responder a través de los proyectos los requerimientos solicitados, es bueno ser multidisciplinario y conocer sobre otras áreas pero es mejor enfocarse en lo más importante que hay para mostrar.

Entonces cuando utilizamos esta herramienta es una buena práctica recopilar información, ver referencias para crear el portafolio y diseñar soluciones que muestren de la mejor forma los proyectos que se han realizado. Hacer un test del portafolio es una buena forma de saber si cumple o no con los objetivos propuestos, de lo contrario se deberá ajustar el diseño.

Finalmente, la gestión del portafolio de proyectos por la complejidad que presenta requiere de madurez organizacional, capital humano altamente competente y una focalización sistemática en proyectos, de lo contrario los beneficios tendrán poco o nulo impacto en los objetivos estratégicos de la organización.

## Herramientas que se pueden utilizar para la creación del portafolio de productos

Existen muchas herramientas que se pueden utilizar para la creación de un portafolio, por ejemplo las herramientas de diseño donde están los software como adobe illustrator, indesign, etc. Y también plataformas web que permiten almacenar y diseñar gratuitamente un espacio para los proyectos como Behance y GitHub, unas de las plataformas más conocidas para portafolios online.

- **GitHub**, es un sitio web y un servicio en la nube que ayuda a los desarrolladores a almacenar y administrar su código, al igual que llevar un registro y control de cualquier cambio sobre este código.
- **Hosting**, es el espacio donde se aloja una página web para que pueda ser visualizado por cualquier usuario a través de un dominio mientras navega por Internet.
- **Página web personal**, es un sitio que alojamos en la web donde podemos exhibir un portafolio, productos o servicios.

- **Youtube**

Si tu trabajo está direccionado a la realización de animaciones en 2D o 3D, la edición de videos o la cinematografía, la mejor opción para mostrar tu trabajo no es un portafolio digital, sino un reel audiovisual o un demo reel que puede ser exhibido a través de youtube u otras plataformas multimedia que permitan reproducir el contenido. Un reel audiovisual, como su nombre lo indica, es la carta de presentación de tus proyectos a través de un clip que contenga audio y video. La idea de realizar un reel es que extraigas fragmentos de tus proyectos más importantes y los recopiles en una sola pieza.

Recuerda que tu portafolio profesional debe ir acompañado de un buen currículum que, además, certifique las habilidades con las que cuentas.

Dentro de las herramientas remota que hay para crear portafolio tenemos las siguientes:

- **Adobe Illustrator**, es un editor de gráficos vectoriales, que permite crear ilustraciones y editar vectores entre otras cosas, es una buena herramienta para diseñar un portafolio ya que su versatilidad y curva de aprendizaje es bastante amigable.
- **Adobe Indesign**, es una aplicación para la composición digital de páginas, al igual que Adobe Illustrator, permite trabajar con diseños vectoriales para proyectos que están dirigidos hacia el área editorial.
- **Adobe XD**, es un editor de gráficos vectoriales, que permite crear prototipos interactivos que pueden a su vez ser testeados por usuarios y entregar gran información en el procesos de la arquitectura de un sitio.
- **Figma**, es un editor de gráficos vectorial y una herramienta de generación de prototipos, al igual que Adobe XD se pueden generar y testear prototipos pero también se puede trabajar colaborativamente, esta herramienta está principalmente basada en la web.

Las herramientas o plataformas que permiten crear y gestionar un portafolio online son las siguientes:

- **Behance**, es una red de sitios y servicios especializada en la autopromoción, incluyendo consultoría y sitios de portafolio en línea. Es propiedad de Adobe
- **Adobe Portafolio**, es una aplicación para exhibir el trabajo creativo. con diversos diseños creados con una cartera en mente para cualquier campo creativo: arte, ilustración, fotografía, diseño gráfico, moda, arquitectura, gráficos animados, diseño web, etc.
- **Carbonmade** permite elegir un template o plantilla de estilo minimalista, que te inspira para subir tus trabajos.
- **Dribbble**, tiene lo mejor de Behance y Carbonmade, es una comunidad para diseñadores, ilustradores, tipógrafos y todos los profesionales del diseño gráfico que te puedas imaginar
- **Wordpress.org**, es más que un sitio web, ya que te permite crear tu propia plataforma con ayuda de su interfaz y mostrar tus proyectos con un dominio y hosting.

## Referencias:

<https://blog.ida.cl/experiencia-de-usuario/construyendo-un-portafolio-ux/>  
<https://es.linkedin.com/pulse/20140703020130-102243624-portafolio-de-proyectos-diferencias-conceptuales-y-su-administraci%C3%B3n>

<https://neoattack.com/neowiki/portafolio/>  
<https://www.crehana.com/cl/blog/desarrollo-web/5-webs-para-crear-tu-primer-portafolio-profesional/>

<https://kinsta.com/es/base-de-conocimiento/que-es-github/>

<https://economipedia.com/definiciones/hosting.html>



## 2.- Finalización del proyecto

### Revisión del producto construido a lo largo del curso

El cuerpo del portafolio está constituido por todos aquellos temas centrales desarrollados a lo largo del curso en cada uno de los módulos. Por tema central, se comprenden todos aquellos conceptos, contenidos y habilidades que el participante adquiere al momento de ejecutar una determinada tarea. Se relacionan con los aprendizajes esperados de los objetivos de cada módulo.

La evidencia de los temas centrales son todos aquellos ejercicios anexados en el portafolio y que concluyen en la generación de una respuesta a la investigación de proyecto alojado en la plataforma digital establecida por el facilitador. Este proyecto viene a ser el producto final y el elemento principal de todo el trabajo desarrollado.

Punto de partida del módulo: Dado que el portafolio es un proceso de evaluación progresiva que pretende que el participante tome una actitud proactiva frente a su aprendizaje, requiere que el facilitador declare desde un comienzo las estrategias evaluativas, al igual que debe establecer las actividades que serán parte del portafolio.

Además, debe procurar que los participantes compartan sus portafolios, de esta manera se establece una rutina de aprendizaje colaborativo, puesto que los participantes podrán revisar las decisiones y análisis que han hecho sus compañeros para ejercicios que están en común. Estas elecciones permiten evidenciar estrategias variadas de aplicación de los mismos contenidos y habilidades desarrollados durante el curso.

El participante en su portafolio debe anexar los ejercicios prácticos que determine el facilitador, manteniendo un directorio ordenado. Para ello deberá:

- En primer lugar, realizar su proyecto según las instrucciones de cada ejercicio
- En segundo lugar, debe producir un documento que contenga los siguientes aspectos:

Módulo [Número del Módulo]	[Nombre del trabajo]
----------------------------	----------------------

Objetivo	[Objetivo declarado, propósito]
Actividad	[Descripción del trabajo que debe ser realizado]
Análisis y reflexión del proceso de resolución	[Descripción del proceso que realizó el participante para llevar a cabo la resolución de la tarea, explicando el porqué de las decisiones tomadas]

- En tercer lugar, debe alojar estos elementos en su directorio dentro de la plataforma digital establecida por el facilitador. De esta manera, cada vez que requiera analizar y revisar algo, puede acceder a su plataforma y revisar lo requerido.

Por lo tanto, el participante debe procurar mantener un orden dentro de su directorio que favorezca la búsqueda de materiales, además, de permitir un análisis global de sus aprendizajes mediante la visualización de una totalidad de trabajos realizados durante el curso.

### **Depuración y mejora del producto**

Establecer un hilo conductor que permita alinear los proyectos según las categorías y adaptar los proyectos en pos de entregar un proyecto mejorado y uniforme, cuando ya establecemos el orden del portafolio con los proyectos que utilizaremos, debemos realizar ajustes con el fin de mejorar el resultado final del portafolio, en cuanto a diseño y contenido del mismo.

### **Feedback y retroalimentación**

El portafolio será evaluado con una lista de cotejo. Esta busca evidenciar tanto el correcto almacenamiento de los trabajos en el directorio como la ejecución de los ejercicios, su reflexión y análisis y la síntesis de los aprendizajes por parte de los participantes.

En esta etapa podemos realizar testeo del prototipo del portafolio y basar la estructura del mismo según los siguientes criterios de evaluación:

Los elementos que debe contener el portafolio son los siguientes:

## Estructura

- Introducción: Presenta una introducción declarando los objetivos del portafolio
- Ejercicios prácticos: Presentar un directorio ordenado correctamente.
- Síntesis del aprendizaje: Identifica claramente cada uno de los ejercicios.

## Contenido

- Actividades o trabajos: Identifica las fases con el objetivo, descripción y análisis del proceso de trabajo.
- Síntesis del aprendizaje: Analiza y argumenta mediante un proceso metacognitivo acerca de los aprendizajes y habilidades desarrollados durante el curso.

## Almacenamiento

- Plataforma digital: Utiliza y publica el portafolio de manera íntegra en plataforma designada.

## Ajustes finales y cierre de entregable

Cuando ya pasamos todas las etapas anteriores ya es hora de finalizar y tomar las últimas decisiones para cerrar el portafolio, para esto debemos considerar los últimos ajustes del trabajo tomando en cuenta el feedback y la retroalimentación del docente, establecer el formato de envío y la visualización declarando las herramientas o plataforma utilizada.

## **Herramientas para la implementación de un portafolio.**

### **Utilización de GitHub para crear un portafolio (\*)**

#### **Qué es GitHub**

Es un sitio web y un servicio en la nube que ayuda a los desarrolladores a almacenar y administrar su código, al igual que llevar un registro y control de cualquier cambio sobre este código. Para entender exactamente qué es GitHub, primero usted necesita conocer los dos principios que lo conectan:

- Control de versión, ayuda a los desarrolladores a llevar un registro y administrar cualquier cambio en el código del proyecto de software. A medida que crece este proyecto, la versión de control se vuelve esencial.
- Git, es un sistema de control específico de versión de fuente abierta creada por Linus Torvalds en el año 2005. Específicamente, Git es un sistema de control de versión distribuida, lo que quiere decir que la base del código entero y su historial se encuentran disponibles en la computadora de todo desarrollador, lo cual permite un fácil acceso a las bifurcaciones y fusiones.

#### **Características y operaciones básicas de Github**

#### **Buenas prácticas para tener un portafolio atractivo en GitHub**

#### **¿Qué es Behance?**

## **Behance**

Behance, parte de Adobe, es una red de sitios y servicios especializada en la autopromoción, incluyendo consultoría y sitios de portafolio en línea, es la plataforma en línea líder para exhibir y descubrir trabajos creativos.

La participación en Behance es gratuita y no hay restricciones en la cantidad de proyectos que un miembro puede crear. Tampoco hay límite en la cantidad de imágenes / medios que los usuarios pueden cargar.

#### **Características y operaciones básicas de Behance**

Después de unirse a Behance, debes personalizar tu perfil, es recomendable proporcionar tanta información como te sientas cómodo compartiendo, tienes la posibilidad de tener una foto de perfil y una foto de banner para darle tu personalidad al sitio.

Crear **Moodboard** es una excelente manera de familiarizarse con Behance, de esta manera puedes navegar por los distintos proyectos del sitio e ir crear un muro con los proyectos que más te inspiren.

Cuando creas un proyecto de tu trabajo, es fácil querer utilizar la mayor cantidad de recursos, pero en muchos casos menos es más. Se sugiere limitar los proyectos a tus mejores imágenes que muestran el tipo de trabajo que quieres seguir haciendo. De esta forma, dejas en claro a los espectadores de tu perfil, ya sean miembros que quieran seguir tu trabajo o reclutadores, el tipo de trabajo que te destaca o te apasiona.

### **Buenas prácticas para tener un portafolio atractivo en Behance**

Para comenzar a promocionar tu trabajo, inicialmente el sitio web Behance puede ser una buena opción, es una plataforma creativa por excelencia. A continuación veremos 4 criterios que debemos tener en cuenta a la hora de diseñar tu portafolio.

**Pública solo contenido de calidad**, es normal sentir la necesidad de publicar todos tus proyectos pero hay tener un ojo crítico sobre lo que quieres enseñar, ya que dependiendo del tipo del trabajo al que estés postulando va a depender que los proyectos que sirvan para demostrar tu experiencia en el área, por esto debes seleccionar los proyectos más acabados y que mejor representen tu forma de trabajar.

**La importancia de la extensión del proyecto**, es necesario mostrar la extensión del proyecto o los procesos que tuviste que pasar para llevar a cabo tus proyectos, en ese sentido demostrar los pasos que seguiste y herramientas que utilizaste hasta llegar al resultado final tiende a ser mejor que solo mostrar los resultados. Lo ideal es que cada proyecto contenga entre 6 y 20 imágenes, un gran scroll que cuente desde el briefing al resultado final pasando por el proceso creativo y sus diferentes aplicaciones.

**La portada es la clave para el éxito**, al dedicar el tiempo a diseñar el contenido de tu proyecto, es importante preocuparse de la portada ya que corres el riesgo de que tu proyecto pase desapercibido. Hay que pensar que



la portada es la cara de tu proyecto es lo primero que se verá entre cientos de proyectos.

**Explora otros portafolios**, el networking es una actitud que todo creativo debe entrenar, Behance es una red perfecta para explorar el trabajo de diseñadores de todo el mundo. Lo interesante de observar otros portafolios es que se puede aprender mucho de otras disciplinas y sirven también como una fuente de inspiración.

### **Behance dedicado y exclusivo a proyectos de UX.**

Behance es el sitio web perfecto para publicar proyectos de UX, ya que su interfaz es bastante amigable y su curva de aprendizaje es sencilla. Permite crear portafolios dinámicos, personas creativas de todo el mundo se unen a behance, ingresando a una comunidad creada con su trabajo original y procesos creativos, lo que surge como una fuente de inspiración y una guía de diseño a la hora de crear un portafolio.

Cuando hablamos de proyectos UX/UI, se debe considerar que un buen portafolio se comprende no solo por el trabajo finalizado, si no también por los procesos y herramientas que se utilizaron para llegar a dichos resultados. En ux generalmente se utilizan varias herramientas de investigación de usuario que evidencia la capacidad del diseñador en observar y reconocer los resultados más significativos que definen el rumbo del proyecto, para esto crear infografías con resultados de encuestas o entrevistas aplicadas, la creación de perfiles de usuarios y procesos de maquetación son fundamentales para declarar las competencias del área.

Dentro del curso debemos tener las siguientes consideraciones para estructurar el portafolio

- El objetivo del portafolio es evidenciar la adquisición de aprendizajes y competencias mediante la recopilación de trabajos realizados durante el curso. El logro de este objetivo depende de una serie de pasos:
- En primer lugar, el participante debe efectuar los trabajos, actividades y ejercicios prácticos propuestos de manera correcta. Esto hace referencia a mantener en el portafolio todos aquellos ejercicios requeridos por el facilitador.



## Qué es un hosting

Alojamiento de tu producto en un servidor

Otro camino para desarrollar un portafolio es la construcción de un sitio web propio, que contenga como vimos tus proyectos más destacados, para esto es necesario que conozcas los hosting

### ¿Qué es un hosting?

Es un servicio de alojamiento para sitios web. El hosting web aloja los contenidos de tu web y tu correo electrónico para que puedan ser visitados en todo momento desde cualquier dispositivo conectado a Internet. Es el hosting quien aloja el contenido del sitio web.

### Servicios gratuitos de hosting

Por lo general el servicio de hosting es de pago, sin embargo, también existen opciones gratuitas un poco más limitadas pero que para el caso de un portafolio son efectivas. Entre los servicios gratuitos más destacados del mercado encontramos:

- **Freehost:** Es una opción con un panel de control muy fácil de usar, además cuenta con plantillas de sitios web.
- **BlueHost:** Este hosting viene con wordpress instalado automáticamente lo cual permite hacer más fácil la gestión del contenido.
- **LucusHost:** Ofrece 1 GB de espacio de almacenamiento, 5 GB de transferencia mensual y soporte técnico 24 horas al día. Además de tecnología PHP

## Cómo alojar un proyecto UX/UI

El primer paso para alojar tu proyecto es conseguir el hosting que se adecue o consideres más fácil de usar, considera las opciones disponibles e investiga. Cuando lo tengas tienes al menos 2 métodos para subir tus archivos del proyecto al sitio web:

### Administradores de archivos

Un administrador de archivos es un programa informático que proporciona una interfaz de usuario para administrar archivos y directorios. El más popular de los administradores de archivo, y que viene en la mayoría de los hosting, es Cpanel. El Administrador de archivos de cPanel es una herramienta muy útil que provee una interfaz web que le permite administrar (añadir, borrar, ver y modificar) archivos dentro del hosting.



Vista del administrador de archivos Cpanel

Para cargar o subir un archivo usando un Administrador de archivos, debes seguir los siguientes pasos:

1. **Abra el administrador de archivos** y navegue a la carpeta en la cual desea cargar el archivo —o archivos—.
2. Haga clic en el **ícono Cargar** desde la barra de herramientas del administrador de archivos.
3. Se abrirá una nueva ventana. **Haga clic en el botón Seleccionar archivo y ubique el archivo que desea cargar** desde su máquina local.
4. **Seleccione el archivo y haga clic en Abrir.**
5. Debes asegurarte que todos tus archivos estén en `public_html`, para que estén visibles en tu web.

Una barra de estatus indicará cuando el proceso se complete (al llegar al 100%).

**Si tienes maquetado el portafolio en HTML y CSS, te conviene este método de administradores de archivos.**

## Plugins de WordPress

Si no tienes tu portafolio maquetado en HTML y CSS conviene buscar un hosting que incluya la opción de integrar WordPress, este gestor de contenidos te permite crear diferentes bloques de contenido para mostrar tus proyectos en tu portafolio web.

## Utilización de youtube en el mundo del desarrollo de productos digitales

Además del portafolio para el perfil de un UX/UI, puede ser provechoso incursionar en la realización de contenido de valor en distintos formatos, cuando decimos contenido de valor nos referimos a contenido que puede ser útil para una comunidad interesada en el tema. Al trabajar con usuarios, todos los proyectos significan un aprendizaje por eso puede ser interesante que puedas comunicarlos. Incluso si aún no tienes proyectos reales, los proyectos ficticios también involucran aprendizaje.

Un formato muy efectivo para realizar este contenido son los videos tutoriales, donde se pueden mostrar distintos procesos o herramientas.

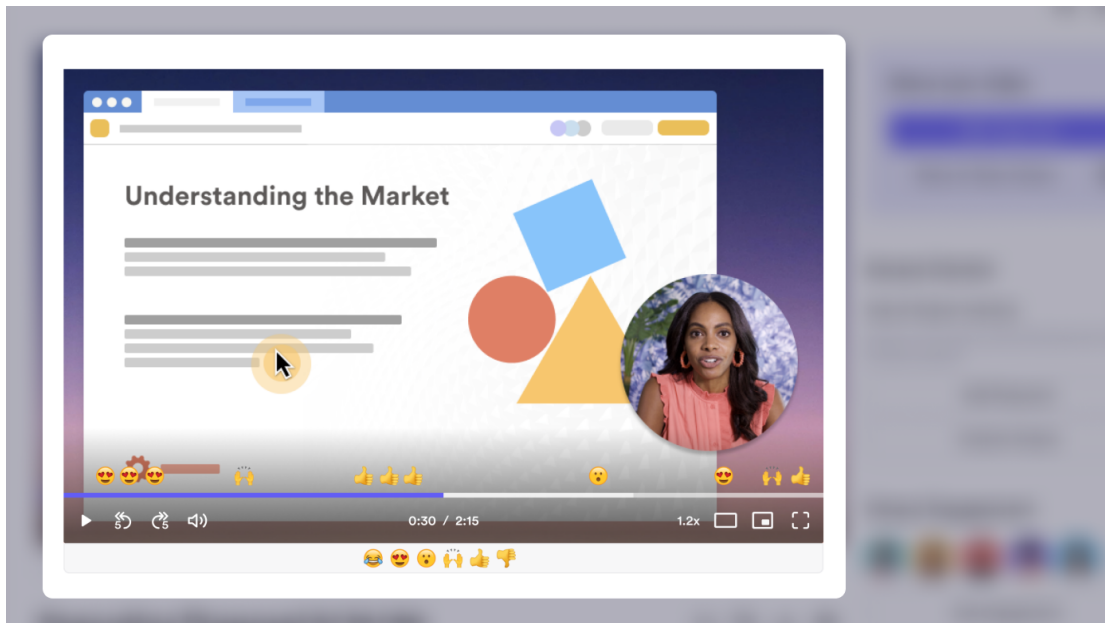
## Herramientas gratuitas para la confección de un video técnico

Una gran herramienta para la confección de videos de todo tipo y material que enganche es **Loom**, es una herramienta gratuita para grabar contenido de tu pantalla de forma fácil, literalmente a un clic.

Las ventajas de esta herramienta son muchas, entre las que más destacamos:

- Es fácil de usar
- Es multiplataforma, cuenta con versión online y descargable para tu escritorio
- Es multifuncional, pues permite la opción de grabar toda la pantalla o el espacio que determines
- Permite mostrar el rostro y grabar pantalla, lo cual genera videotutoriales en los que se puede ver al relator, una función que te permite mostrar tus capacidades comunicativa
- Es gratuita





Vista de un video desarrollado en Loom

<https://www.loom.com/>

## Cómo alojar un proyecto Front-End

Como subir mi web a la nube usando Heroku

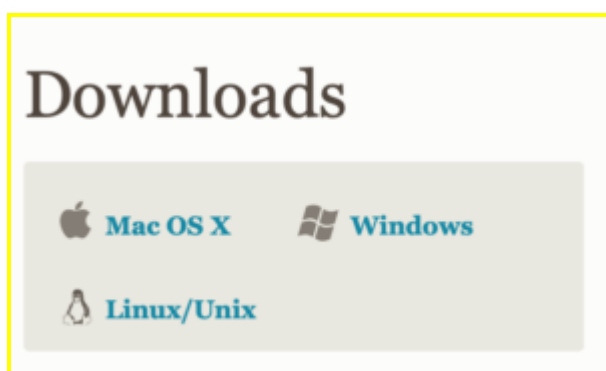
Pasos para subir tu web a la nube usando Heroku

1. Instalación de las herramientas a utilizar

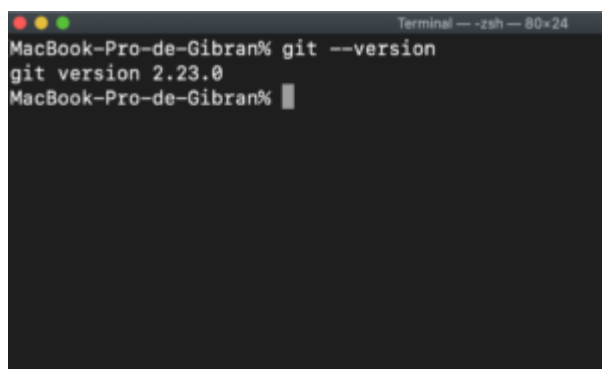
Utilizaremos Git y la interfaz de comandos de Heroku. También se requerirá crear una cuenta en Heroku.

Git

Descarga Git en su página de descargas: <https://git-scm.com/downloads>



Una vez descargado, instálalo, y prueba abriendo una ventana de comandos y escribiendo:



## Heroku Cli

Descarga la interfaz de comandos de Heroku en la siguiente página:

<https://devcenter.heroku.com/articles/heroku-cli>



## Download and install



The Heroku CLI requires Git, the popular version control system. If you don't already have Git installed, complete the following before installing the CLI:

- [Git installation](#)
- [First-time Git setup](#)

### 🍏 macOS

Download the installer

Also available via Homebrew:

```
$ brew tap heroku/brew && brew install heroku
```

### 🪟 Windows

Download the appropriate installer for your Windows installation:

64-bit installer


32-bit installer

### 🐧 Ubuntu 16+

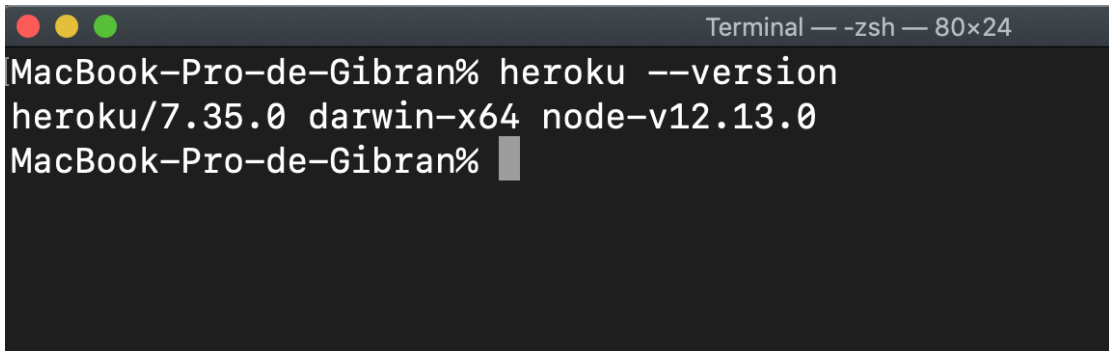
Run the following from your terminal:

```
$ sudo snap install --classic heroku
```

Una vez descargado instálalo, y prueba abriendo tu ventana de comandos de tu sistema operativo y escribiendo el comando:



```
heroku --version
```



```
Terminal — -zsh — 80x24
MacBook-Pro-de-Gibran% heroku --version
heroku/7.35.0 darwin-x64 node-v12.13.0
MacBook-Pro-de-Gibran%
```

### Cuenta en Heroku

Crea una cuenta en Heroku.

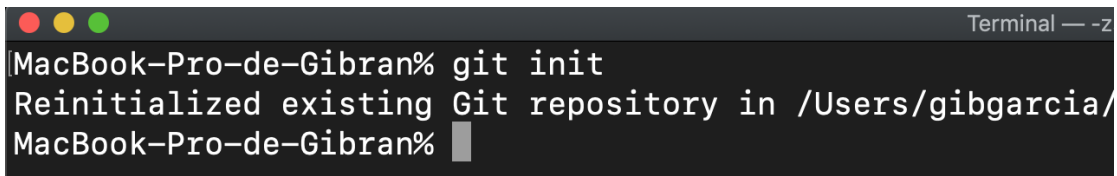
<https://signup.heroku.com>

## 2. Generar un repositorio local en Git

Abre una ventana de comandos (CMD o Terminal) y ubícate en la carpeta donde está tu proyecto. Importante: cambia la extensión del archivo .html por .php.

Generar un repositorio local

Escribe el comando



```
Terminal — -z
MacBook-Pro-de-Gibran% git init
Reinitialized existing Git repository in /Users/gibgarcia/
MacBook-Pro-de-Gibran%
```

Ahora agregaremos los archivos de nuestro proyecto al repositorio

```
Terminal — -z
MacBook-Pro-de-Gibran% git add .
MacBook-Pro-de-Gibran%
```

Hacemos un commit para confirmar los cambios que acabamos de realizar

```
Terminal — -z
MacBook-Pro-de-Gibran% git commit -m "ProyectoFGGC"
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
MacBook-Pro-de-Gibran%
```

### 3. Crear una app en Heroku

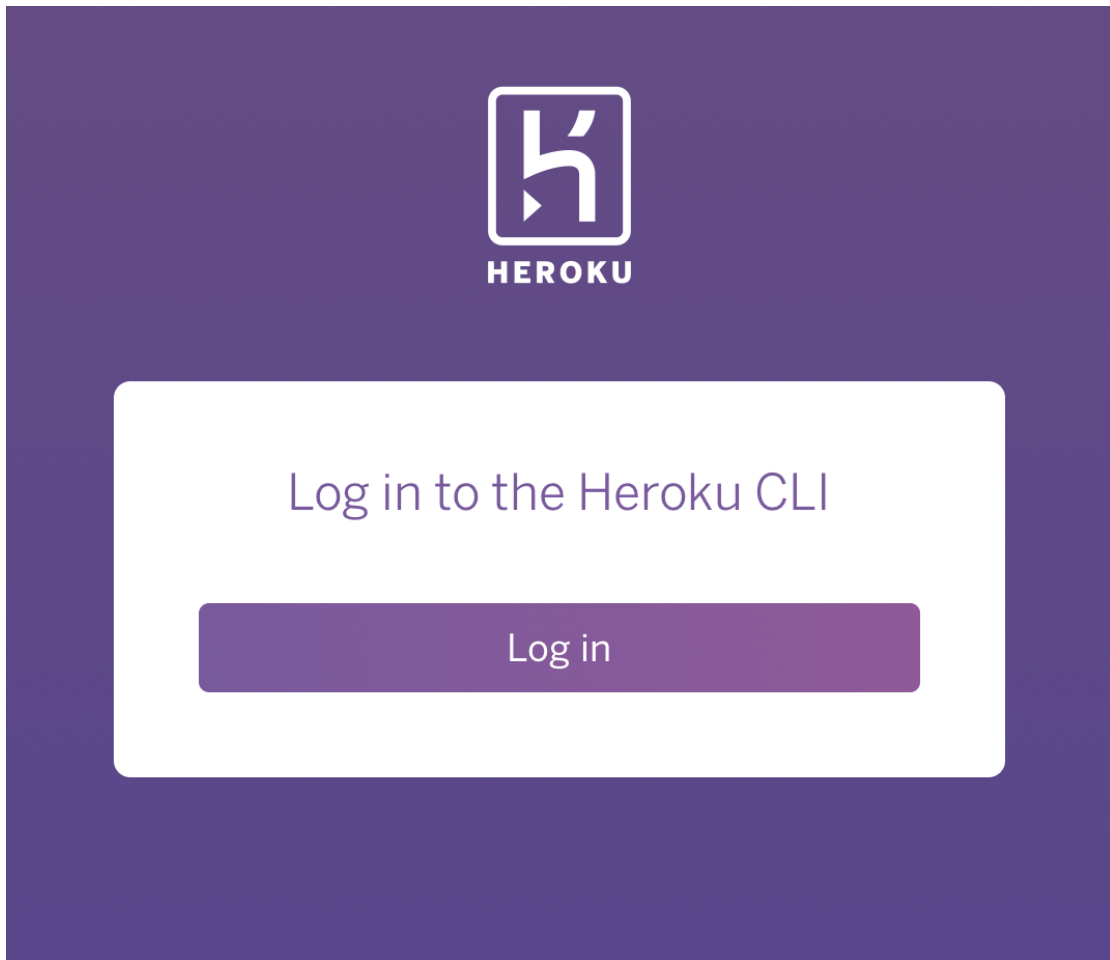
Hacemos login en Heroku

```
heroku login

Terminal — node • heroku login — 80x24
MacBook-Pro-de-Gibran% heroku login
heroku: Press any key to open up the browser to login or q to exit:
```

Te pedirá ir a la web de Heroku para autenticarse. Deberás proporcionar tu nombre de usuario y contraseña que obtuviste cuando creaste tu cuenta en Heroku.





Crear una app en Heroku

```
heroku create
```

```
MacBook-Pro-de-Gibran% heroku create
Creating app... done, ⬢ salty-taiga-06242
https://salty-taiga-06242.herokuapp.com/ | https://git.heroku.com/salty-taiga-06242.git
MacBook-Pro-de-Gibran% 
```

Heroku creará una app utilizando un nombre “aleatorio”. Toma nota de la url que generó, pues en ese punto se colgará nuestra web.

Subimos nuestro proyecto a Heroku

```
git push heroku master
```

```
MacBook-Pro-de-Gibran% git push heroku master
Enumerating objects: 39, done.
Counting objects: 100% (39/39), done.
Delta compression using up to 4 threads
Compressing objects: 100% (39/39), done.
Writing objects: 100% (39/39), 4.06 MiB | 16.00 KiB/s, done.
Total 39 (delta 4), reused 0 (delta 0)
```

Se demorará el tiempo necesario de acuerdo a la cantidad y tamaño de nuestros archivos. Una vez finalizado éste proceso podrás ir a la URL generada en el paso anterior usando tu navegador web preferido. Felicidades, ya tienes tu sitio en línea.

## Cómo alojar un proyecto Full Stack Java

La tecnología JSF (JavaServer Faces), es usada por muchos desarrolladores de páginas web que cuyo propósito es realizar páginas web de una manera dinámica e interactiva.

JSP (JavaServer Faces), es una librería de JSF la cual permitirá realizar páginas JSP utilizando un propio etiquetado como también el uso de html.

Hoy en día el Internet no cuenta con un importante número de servidores que provean de hosting (alojamiento) a páginas web desarrolladas en JSP.

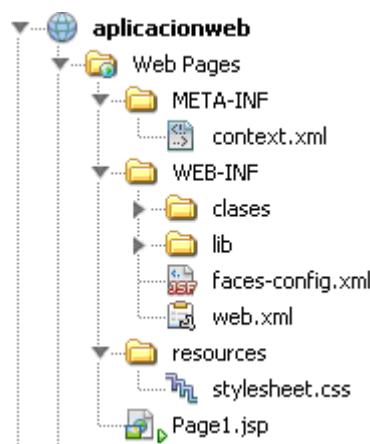
Para esto necesitamos disponer de un archivo .War

### GENERALIDADES

- Definición Archivo War. Es nuestra aplicación web java empaquetada o comprimida lista para ser desplegada en cualquiera de los servidores que brindan servicios para este tipo de aplicaciones web.
- Ventaja. La gran ventaja de los WAR es que te permiten olvidarte de los detalles de configuración, siempre que tu entorno de programación lo automatice, que es lo normal.

### ESTRUCTURA DE UN ARCHIVO WAR

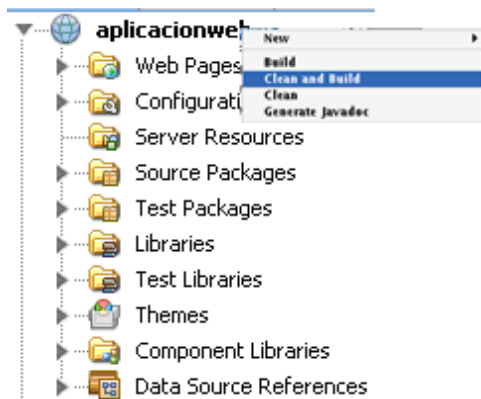
- La estructura de un archivo war es la mismo que usa la aplicación web. En la figura podemos ver la estructura de directorios.



## FORMAS DE CREAR UN ARCHIVO WAR

### \* Usando un IDE

- Después de crear un proyecto con éxito el IDE (Entorno de Desarrollo Integrado) nos genera el archivo .war automáticamente.
- Para entender de mejor manera la creación del archivo war de una aplicación web usamos como IDE netbeans 6.0.
- Hacer clic derecho sobre el nombre de la aplicación web (aplicacionweb) y luego Clean and Build.



- Esto genera una carpeta que contiene el archivo war de la aplicación, como se puede apreciar en la figura.

```
Created dir: C:\Archivos de programa\Apache Software Foundation\Apache Tomcat 6.0.14\webapps\aplicacionweb\dist
Building jar: C:\Archivos de programa\Apache Software Foundation\Apache Tomcat 6.0.14\webapps\aplicacionweb\dist\aplicacionweb.war
do-dist:
dist:
BUILD SUCCESSFUL (total time: 13 seconds)
```

### \* Usando el DOS de Windows

- Ir al directorio raíz de la aplicación mediante comandos desde el DOS.

```
C:\Archivos de programa\Apache Software Foundation\Apache Tomcat 6.0.14\webapps\aplicacionweb>_
```

- Seguidamente tecleamos el siguiente comando:

jar cv0f aplicacionweb.war .

- Observa el punto obligatorio que hay al final de la línea; le dice al programa jar que archive el directorio actual.

```
manifest agregado
agregando: index.html <entr
agregando: WEB-INF/ <entra
agregando: WEB-INF/classes/
agregando: WEB-INF/classes/
do 0%>
agregando: WEB-INF/classes/
o 0%>
agregando: WEB-INF/web.xml
```

- El comando jar anterior crea un fichero llamado aplicacionweb.war



## 2. Comprar un dominio

# Domain Names

## .COMs starting at \$9.99!\*

- Las Extensiones de Dominios es la parte final del nombre de un dominio como: .com , .net , .org , entre otros. Se debe escoger el tipo de dominio que más le convenga dependiendo del tipo de empresa o negocio.

- Generalmente no requerimos ningún tipo de documento para registrar un dominio, a excepción de los dominios cuya extensión terminan en .edu.ec , .gob.ec .

- La empresa SECUREPAYNET, viene ofreciendo el servicio de dominios, previo a la compra de un dominio debemos saber si se encuentra disponible para la cual hicimos una prueba y el resultado es el que se muestra en la figura.

**Start Your Domain Name Search Here!**

Enter a Domain Name:

www.  .com

- El dominio ingresado esta disponible en diferentes extensiones.



**CVUCIS.COM is available!**

Check the domain names that you would like to register below.

► [Why you need multiple domains](#)

	.com*	.info*	.mobi*	.net*	.org*	.tv
Select All	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	\$9.99/yr	\$15.00/yr	\$14.99	\$9.99/yr	\$9.99/yr	\$44.99/yr





- El link de la empresa que provee este servicio es:  
<http://www.securepaynet.net/>

### 3. Comprar un hosting

- Una vez que se tiene hecha tu aplicación web java, necesitas enviarla a un servidor que esté permanentemente conectado a Internet para que cualquier internauta pueda verla. Este tipo de servicio se llama hosting o alojamiento es el lugar donde se va encontrar la aplicación web.



- La empresa HOSTLGNITION ofrece tres niveles de jsp & Servlets de Java ofreciendo las soluciones que son diseñadas para complacer los requisitos de aplicación únicos de clientes.

#### NIVEL UNO:

- Compartir la máquina virtual de java con otros usuarios.
- Disponible sobre StarterFire y hacia arriba.
- Recomendada para: scripts / servlets del PSJ simples sitios web personales
- Las características estándares:
  - \* Apache como servidor de la aplicación y Tomcat como motor de servlets.
  - \* Java Development Kit 1.6.0 para compilar y ejecutar los programas.
  - \* Acceso completo al directorio WEB-INF
  - \* web.xml para la configuración de los servlets mapping, etc
  - \* Acceder a las clases java instaladas (JDBC) Disponibilidad 24 / 7
  - \* Protocolo de transferencia de archivos (FTP), gestión de archivos
- En el nivel 1 JSP/Servlets soporta cisco planes estándares de

alojamiento web como se muestra en la figura siguiente:

JSP/Servlets Hosting	StarterFire	LiteFire	ProFire	EliteFire	MegaFire
<b>Installed Software</b>	Tomcat: <b>5.5.x</b>				
	Java Development Kit: <b>1.6.0</b>				
	Apache: <b>2.2.x</b>				
	MySQL: <b>5.0.x</b>				
<b>Level 1</b>	\$1.00 per month	FREE			
<b>Web Space</b>	6GB	16GB	40GB	80GB	160GB
<b>Bandwidth</b>	200GB	400GB	800GB	1200GB	2000GB
<b>Other Development Features</b>	PHP, Perl, MySQL, PostgreSQL, Python, Ruby on Rails, and more.				
<b>Monthly Pricing</b>	\$4.95	\$9.95	\$17.95	\$29.95	\$59.95

- El link de la empresa que provee este servicio es:  
<http://www.hostignition.com/>

#### 4. Transferir la aplicación

##### GENERALIDADES

- Para subir la aplicación web java al servidor de alojamiento utilizaremos FTP (Protocolo de Transferencia de Archivos).
- Utilizaremos un cliente FTP (FileZilla) que básicamente es quien realiza la llamada de conexión al servidor.
- Por defecto FTP utiliza el puerto 21. Para enviar las órdenes del cliente al servidor o para la transferencia de archivos.

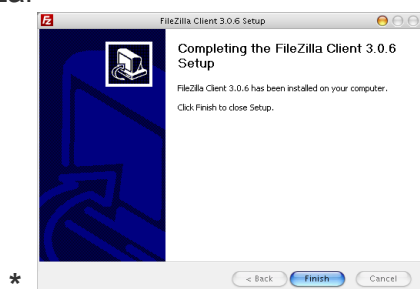
##### MANERAS DE SUBIR ARCHIVOS FTP AL SERVIDOR

- \* Usando un cliente FTP.
- \*\* FileZilla v 3.0.6 Cliente FTP
  - Tamaño: 2,92 MB
  - Licencia: GNU General Public License (GPL)
  - URL: <http://filezilla.sourceforge.net>
  - Documentación: <http://filezilla.sourceforge.net/documentation/>
  - Descarga:  
[http://prdownloads.sourceforge.net/filezilla/FileZilla\\_2\\_2\\_28\\_setup.exe](http://prdownloads.sourceforge.net/filezilla/FileZilla_2_2_28_setup.exe)

##### \*\* Instalación de FileZilla.

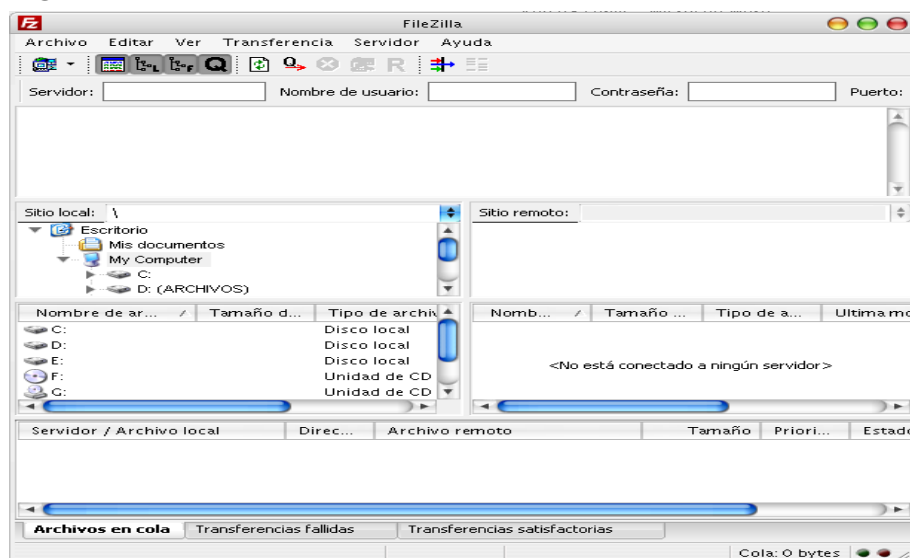
- Una vez descargado el archivo ejecutable procedemos a la instalación.
- Doble click sobre el ejecutable > Aceptar la Licencia > Seleccionar si usaran todos los usuarios de la máquina o solo el actual > Siguiente >

Seleccionar componentes de preferencia todos > Siguiente > Seleccionar el directorio de instalación > Seleccionar el nombre del menú inicio > Instalar > Finalizar



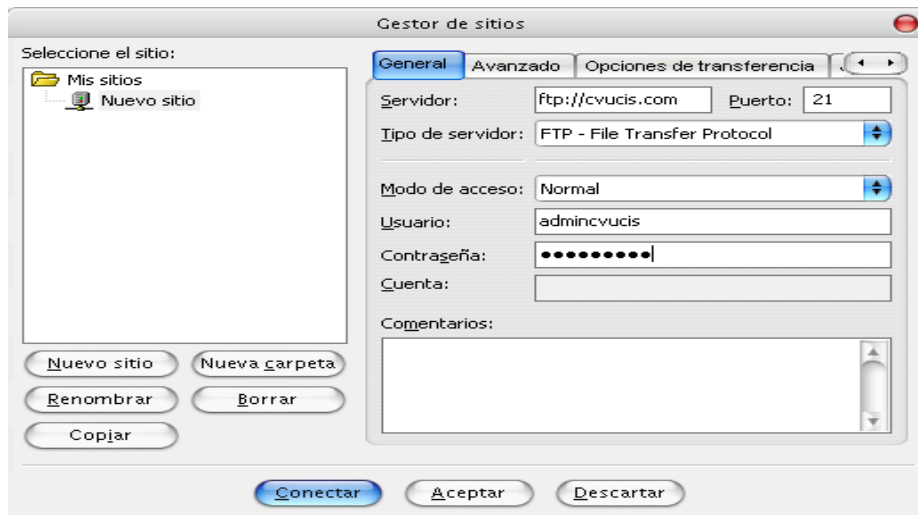
## \*\* Iniciar FileZilla

- Hacer doble click sobre el icono que se encuentra en el escritorio.
  - Cambiar de idioma: Editar > Opciones > Lenguaje > Select Language
- > OK.

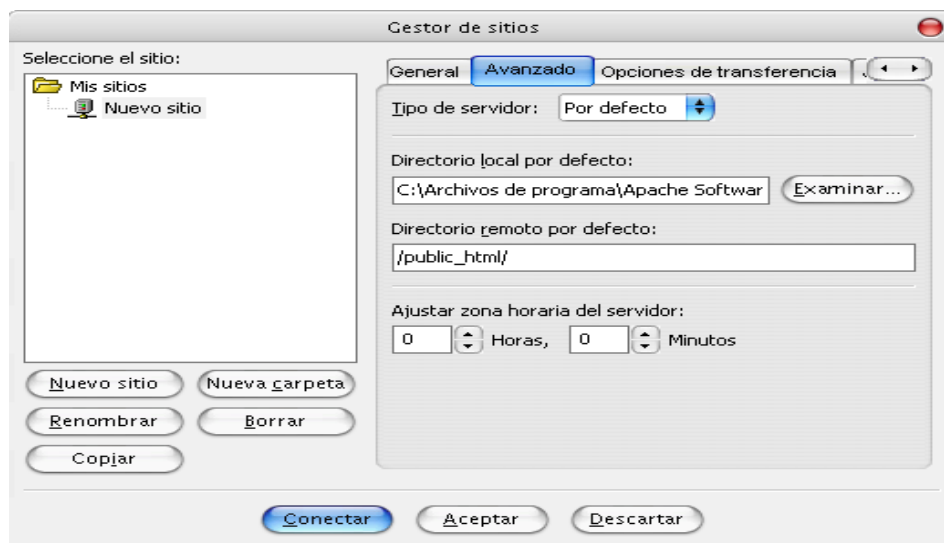


## \*\* Conectarse al Servidor

- Previo al conectarse al servidor debemos disponer de la siguiente información: Nombre del Servidor.- <ftp://tu-dominio.com>, Puerto.-21, Usuario.- Tu nombre de usuario y Contraseña.- Palabra clave.
- Ahora click en Archivo > Gestor de Sitios > Nuevo Sitio > Introducir los datos anteriores para la conexión > Conectar.



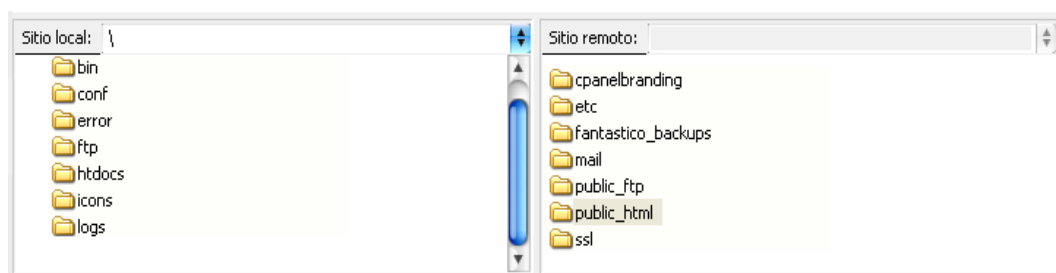
- Click en Avanzado podemos seleccionar el directorio local por defecto y el directorio remoto por defecto.



- Ventana superior.- La consola de salida (output). Muestra los comandos enviados por FileZilla y las respuestas del servidor.



- Ventana Izquierda.- Interfaz gráfica de los contenidos del cliente FTP, es decir, la máquina local.
- Ventana Derecha.- Interfaz gráfica de los contenidos del servidor FTP, es decir la máquina remota.



- Ventana Inferior.- muestra en tiempo real el estado de las transferencias activas o pendientes de envío.

```
E:\apache2triad\htdocs\wperu\tutoria... 23726 --->> /public_html/tutoriales/detectar-ip-del... wperu.cc
Transferring
E:\apache2triad\htdocs\wperu\tutoria... 24158 --->> /public_html/tutoriales/detectar-pais-d... wperu.cc
Transferring
```

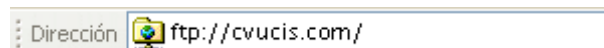
**\*\* Directorio public\_html en el sitio remoto**

- Para subir archivos, seleccione los archivos/carpetas en el sitio local y 'arrastrellos' al directorio apropiado (public\_html) del sitio remoto.

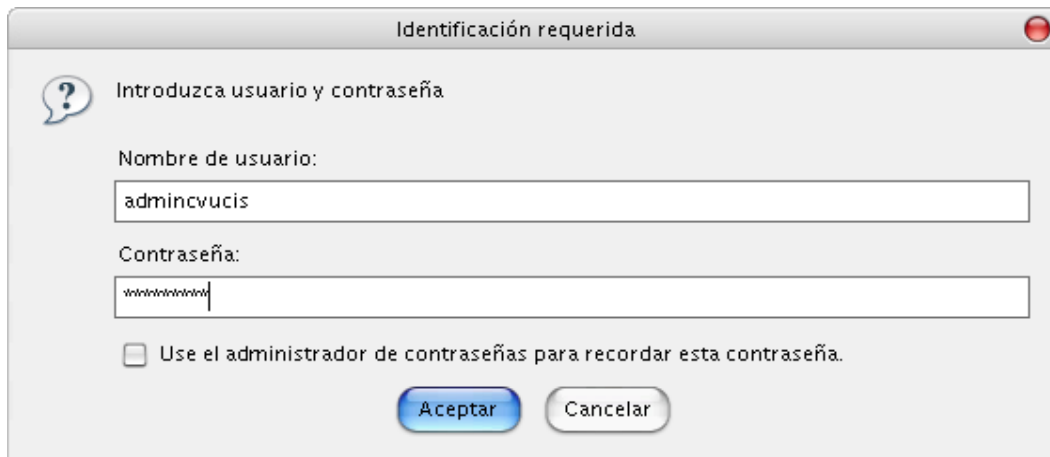
**\* Usando un Navegador Web Mozilla-Firefox v 2.0.0.11**

**\*\* Para tener acceso vía web será necesario: Nombre del Servidor.- ftp://tu-dominio.com, Usuario.- Tu nombre de usuario y Contraseña.- Palabra clave.**

**\*\* Escribir en el navegador Nombre del Servidor luego presionar intro.**



\*\* Se visualiza una pantalla como la siguiente; ingrese su Usuario y Contraseña, luego clic en el botón Aceptar.



\*\* Si todo ha sido exitoso se debe de mostrar una ventana similar a ala siguiente; doble clic en la carpeta public\_html.

\*\* Finalmente, copie todos sus archivos web dentro de la carpeta public\_html; para ello use copy and paste.

## 5. Licencia



Esta obra esta bajo la licencia de Creative Commons en tu eres libre de:

- \* Copiar, distribuir, comunicar y ejecutar públicamente la obra
- \* Hacer obras derivadas Bajo las siguientes condiciones:
- \* Reconocimiento. Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).

Ver una copia en el sitio:

[http://www.creativecommons.org/licenses/by/2.5/es/deed.es\\_CL](http://www.creativecommons.org/licenses/by/2.5/es/deed.es_CL)



## Cómo alojar un proyecto Full Stack JavaScript

Vamos a ver el paso a paso de cómo subir tu aplicación web de Node.js en Heroku, sin importar el framework que uses (por ejemplo, express) de manera fácil, rápida y gratuita.

Un resumen rápido para esto es:

En resumen es:

- Instalar CLI de heroku
- Agregar git a tu proyecto (o clonarlo si ya lo usas)
- Usar **heroku login** para iniciar sesión
- Ejecutar **heroku create nombre\_de\_app** dentro de tu proyecto
- Crear **Procfile** y ajustar puerto en caso de que tu app sea web
- Ejecutar **git push heroku master**
- Probar tu app web

## Login en Heroku

Una vez que tengas la CLI de heroku ejecuta:

**heroku login**

```
root@parzibyte:/# heroku login
(node:6312) SyntaxError Plugin: heroku: C:\Users\usuario\AppData\Local\heroku\version: Unexpected end of JSON input
module: @oclif/config@1.13.2
task: runHook init
plugin: heroku
root: C:\Program Files\heroku\client
See more details with DEBUG=*
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/browser/
Logging in... done
Logged in as
```

Iniciar sesión en Heroku desde la cli de Heroku

Inicia sesión en el navegador, autoriza la app y listo.

## Preparar proyecto

Tu proyecto debe ser un proyecto de Git, ya mencioné que puedes usar Github desktop. Si sabes usar la línea de comandos de git también es válido, el punto es que tu proyecto debe usar git.

Abre la línea de comandos y navega a tu proyecto (`cd nombre_de_proyecto`), después ejecuta:

`heroku create nombre-de-tu-app`

```
root@parzibyte:~# heroku create app-node-parzibyte
(node:7192) SyntaxError Plugin: heroku: C:\Users\usuario\AppData\Local\heroku\version: Unexpected end of JSON input
module: @oclif/config@1.13.2
task: runHook init
plugin: heroku
root: C:\Program Files\heroku\client
See more details with DEBUG=*
Creating • app-node-parzibyte... done
https://app-node-parzibyte.herokuapp.com/ | https://git.heroku.com/app-node-parzibyte.git
```

Eso modificará el repositorio de git para que luego lo puedas publicar en Heroku.

Por otro lado, en tu app debe estar definido el `package.json` pues Heroku leerá las dependencias del mismo e instalará los módulos por ti.

Ahora necesitamos dos cosas:

- El Procfile
- Ajustar el puerto, en caso de que tu app sea una app web

## Procfile en node

Este archivo indica cómo se va a ejecutar tu app de Node en Heroku. Cuando desarrollamos localmente probablemente escribimos:

`node index.js`

Exactamente eso debemos indicar en el **Procfile**, así:

```
1 web: node index.js
```

Procfile hosted with ❤ by GitHub

[view raw](#)

Como puedes ver, es un archivo sin extensión. Indica que el tipo de proceso es web, y para ejecutarse debe invocar a **node index.js**

Haz commit de este archivo como normalmente lo haces.

## Ajustar puerto

En Heroku no vamos a tener cualquier puerto disponible, pues lo provee, digamos, el sistema. El puerto estará en **process.env.PORT** así que si tenemos un código algo así:

```
1 const express = require("express"),
2   path = require("path"),
3   app = express(),
4   puerto = 3000; // <-- este es el puerto local
```

index\_mal.js hosted with ❤ by GitHub

[view raw](#)

Debemos cambiarlo para que, si está definido el puerto en el entorno, use ese en su lugar. Así:

```
1 const express = require("express"),
2   path = require("path"),
3   app = express(),
4   puerto = process.env.PORT || 3000; // Si está definido en el entorno, usarlo. Si no, el 3000
```

index\_ok.js hosted with ❤ by GitHub

[view raw](#)

Fue un ejemplo, en tu código puedes definir el puerto en otro lugar, pero así es en el mío.

También **haz commit** de este cambio, es decir, tu flujo de trabajo no va a cambiar, sigue subiendo tus cambios.

## Ahora sí, publicar

Cuando hayas hecho commit y todo esté listo, ejecuta:

```
git push heroku master
```

Estamos subiendo nuestros cambios a la rama master de heroku. Verás algo así:

```
C:\WINDOWS\system32\cmd.exe
root@parzibyte:/# git push heroku master
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 4 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (14/14), 7.29 KiB | 1.46 MiB/s, done.
Total 14 (delta 2), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:       NPM_CONFIG_LOGLEVEL=error
remote:       NODE_ENV=production
remote:       NODE_MODULES_CACHE=true
remote:       NODE_VERBOSE=false
remote:
remote: -----> Installing binaries
remote:       engines.node (package.json):  unspecified
remote:       engines.npm (package.json):   unspecified (use default)
remote:
remote:       Resolving node version 12.x...
remote:       Downloading and installing node 12.13.1...
remote:       Using default npm version: 6.12.1
remote:
remote: -----> Installing dependencies
remote:       Installing node modules (package.json + package-lock)
remote:       added 50 packages from 37 contributors and audited 126 packages in 1.652s
remote:       found 0 vulnerabilities
remote:
remote: -----> Build
remote:
```

Y al final te dará la URL de tu proyecto:

```
remote: ----> Build succeeded!
remote: ----> Discovering process types
remote:      Procfile declares types -> web
remote:
remote: ----> Compressing...
remote:      Done: 21.3M
remote: ----> Launching...
remote:      Released v4
remote:      https://app-node-parzibyte.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/app-node-parzibyte.git
   1b758e8..f2ba9c1  master -> master

root@parzibyte:/#
```

## Localizar web en donde está la app de Heroku

La cual puedes visitar y estará lista para lo que quieras:

← → ↻ 🏠 🔒 app-node-parzibyte.herokuapp.com/pagina

## Esta plantilla HTML será enviada con Express

### Actualizar app

Si después haces cambios en tu aplicación, haz lo siguiente:

1. Cambia el código, plantilla, archivo necesario
2. Haz commit y sube tus cambios a GitHub como normalmente se hace
3. Después ejecuta (dentro del directorio de tu app) `git push heroku master` y tu app será desplegada

### Conclusión y notas finales

De este modo puedes subir tus apps de Node y Express (u otro framework) a la nube de manera gratuita y fácil. Recuerda que el plan gratuito tiene sus restricciones.

## Cómo alojar un proyecto Python

### Despliegue en Heroku (así como en PythonAnywhere)

Heroku es también gratuito para pequeñas aplicaciones que no tienen muchas visitas, pero un poco más complicado para desplegar.

### El archivo requirements.txt

Si no lo creaste antes, necesitamos crear ahora un archivo requirements.txt para decirle a Heroku qué paquetes de Python necesitan estar instalados en nuestro servidor.

Pero primero, Heroku necesita que nosotros instalemos algunos paquetes nuevos. Ve a la consola con tu virtualenv (entorno virtual) activado y escribe lo siguiente:

```
(myvenv) $ pip install dj-database-url gunicorn whitenoise
```

Cuando finalice la instalación, ve al directorio.djangogirls y ejecuta este comando:

```
(myvenv) $ pip freeze > requirements.txt
```

Esto creará un archivo llamado requirements.txt con una lista de tus paquetes instalados (i.e. bibliotecas de Python que estás usando, por ejemplo Django :)).

**Nota:** pip freeze lista todas las bibliotecas Python instaladas en tu virtualenv, y el > toma la salida de pip freeze y la pone en un archivo. ¡Intenta ejecutar pip freeze sin el > requirements.txt para ver qué pasa!

Abre este archivo y añádele la siguiente línea al final:

```
psycopg2==2.5.4
```

Esta línea se necesita para que tu aplicación funcione en Heroku.



## Procfile

Otro archivo que Heroku necesita es el Procfile. Éste le dice a Heroku qué comandos ejecutar para iniciar nuestro sitio web. Abre el editor de código, crea un archivo llamado Procfile en el directorio.djangogirls y añade esta línea:

```
web: gunicorn mysite.wsgi
```

Esta línea significa que vamos a desplegar una aplicación web, y que lo haremos ejecutando el comando gunicorn mysite.wsgi (gunicorn es como una versión más potente del comando runserver de Django).

Ahora guárdalo. ¡Hecho!

## El archivo runtime.txt

También necesitamos decirle a Heroku qué versión de Python queremos usar. Esto se hace creando un archivo runtime.txt en el directorio.djangogirls usando el comando "nuevo archivo" de tu editor, y poniendo el siguiente texto (¡y nada más!) dentro:

```
python-3.4.2
```

## mysite/local\_settings.py

Como es más restrictivo que PythonAnywhere, Heroku necesita usar una configuración diferente de la que usamos localmente (en nuestro ordenador). Por ejemplo, Heroku quiere usar Postgres mientras que nosotros usamos SQLite. Por eso necesitamos crear un archivo de configuración separado que estará sólo disponible para nuestro entorno local.

Continúa y crea el archivo mysite/local\_settings.py. Debería contener la configuración para DATABASE de tu archivo mysite/settings.py. Así:

```
import os
BASE_DIR = os.path.dirname(os.path.dirname(__file__))

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
```

```
'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),  
}  
}
```

```
DEBUG = True
```

Ahora ¡sólo guárdalo! :)

### **mysite/settings.py**

Otra cosa que necesitamos hacer es modificar el archivo `settings.py` de nuestro sitio web. Abre `mysite/settings.py` en tu editor y añade las siguientes líneas al final del archivo:

```
import dj_database_url  
DATABASES['default'] = dj_database_url.config()  
  
SECURE_PROXY_SSL_HEADER = ('HTTP_X_FORWARDED_PROTO',  
                             'https')  
  
ALLOWED_HOSTS = ['*']  
  
DEBUG = False  
  
try:  
    from .local_settings import *  
except ImportError:  
    pass
```

Esto hará la configuración necesaria para Heroku y también importará tu configuración local si existe el archivo `mysite/local_settings.py`.

Guarda el archivo.

### **mysite/wsgi.py**

Abre el archivo `mysite/wsgi.py` y añade estas líneas al final:

```
from whitenoise.django import DjangoWhiteNoise  
application = DjangoWhiteNoise(application)
```

## Cuenta en Heroku

Necesitas instalar la herramienta Heroku *toolbelt* que puedes encontrar aquí (puedes saltarte la instalación si ya lo has hecho durante la configuración inicial): <https://toolbelt.heroku.com/>

Si ejecutas el programa de instalación de Heroku *toolbelt* en Windows asegúrate de elegir "Custom Installation" cuando te pregunte qué componentes instalar. En la lista de componentes que aparece después añade a la selección "Git and SSH".

En Windows también debes ejecutar la siguiente instrucción para añadir Git y SSH al PATH de tu ventana de comandos: `setx PATH "%PATH%;C:\Program Files\Git\bin"`. Reinicia tu ventana de comandos para habilitar el cambio.

Después de reiniciar tu ventana de comandos, ¡no olvides ir a tu carpeta `djangogirls` otra vez y activar tu entorno virtual!

Crea también, por favor, una cuenta gratuita de Heroku aquí: <https://id.heroku.com/signup/www-home-top>

Luego autentica tu cuenta de Heroku en tu ordenador ejecutando el siguiente comando:

```
$ heroku login
```

En caso de que no tengas una clave SSH, este comando generará una automáticamente. Las claves SSH se necesitan para hacer *push* de código a Heroku, es decir, enviar tu código al servidor.

## Git commit

Heroku usa git para sus despliegues. A diferencia de PythonAnywhere, puedes hacer un *push* a Heroku directamente, sin utilizar Github como intermediario. Pero necesitamos hacer unas pequeñas modificaciones antes.

Abre el archivo llamado `.gitignore` en tu directorio `djangogirls` y añade `local_settings.py` al mismo. Queremos que git ignore `local_settings`, para que se quede en nuestro ordenador local y no sea subido a Heroku.

```
*.pyc
db.sqlite3
myenv
__pycache__
local_settings.py
```

Y hacemos un commit de nuestros cambios

```
$ git status
$ git add -A .
$ git commit -m "archivos adicionales y cambios para Heroku"
```

## Selecciona un nombre para la aplicación

Estamos haciendo nuestro blog disponible en la Web en `[nombre de tu blog].herokuapp.com`, por lo que necesitamos seleccionar un nombre que nadie más esté usando. Este nombre no tiene que estar relacionado con nuestra aplicación blog de Django o con `mysite` ni con nada de lo que hemos creado hasta ahora. El nombre puede ser cualquier cosa que tú quieras, pero Heroku es bastante estricto con respecto a los caracteres que puedes usar: sólo se permiten letras minúsculas simples (no mayúsculas ni acentos), números y guiones (-).

Una vez que hayas pensado en un nombre (quizás algo con tu nombre o tu apodo), ejecuta este comando, reemplazando `djangogirlsblog` con el nombre seleccionado:

```
$ heroku create djangogirlsblog
```

**Nota:** Recuerda reemplazar `djangogirlsblog` con el nombre de tu aplicación en Heroku.

Si no se te ocurre ningún nombre, también puedes ejecutar simplemente:

```
$ heroku create
```

y Heroku elegirá un nombre no usado para ti (probablemente algo como `enigmatic-cove-2527`).

Si en algún momento te apetece cambiar el nombre de tu aplicación en Heroku, puedes hacerlo con este comando (reemplaza `nombre-nuevo` con el nuevo nombre que quieras usar):

```
$ heroku apps:rename nombre-nuevo
```

**Nota:** Recuerda que después de cambiar el nombre de la aplicación, tienes que visitar `[nombre-nuevo].herokuapp.com` para ver tu sitio web.

## ¡Desplegar en Heroku!

Eso fue un montón de configuración e instalación, ¿verdad? ¡Pero sólo necesitas hacerlo una vez! ¡Ahora ya puedes desplegar!

Cuando ejecutaste `heroku create`, automáticamente se añadió el Heroku *remote* para la aplicación en nuestro repositorio. Ahora podemos hacer un simple `git push` para desplegar nuestra aplicación:

```
$ git push heroku master
```

**Nota:** Esto probablemente producirá un *montón* de salida por consola la primera vez que lo ejecutas, ya que Heroku compila e instala `psycopg`. Sabrás que has tenido éxito si ves algo como `https://tunombredeaplicacion.herokuapp.com/` deployed to Heroku hacia el final del texto.

## Visita tu aplicación

Has desplegado tu código en Heroku, y especificado el tipo de proceso en un `Procfile` (nosotros elegimos tipo de proceso `web`). Ahora podemos decirle a Heroku que inicie este proceso web.

Para hacerlo, ejecuta el siguiente comando:

```
$ heroku ps:scale web=1
```

Este comando le dice a Heroku que ejecute solamente una instancia de nuestro proceso web. Ya que nuestra aplicación de blog es bastante simple, no necesitamos demasiada potencia y por tanto está bien ejecutar solamente un proceso. Es posible pedirle a Heroku que ejecute más procesos (por cierto, Heroku le llama a estos procesos "Dynos" así que no te sorprendas si ves ese nombre) pero entonces dejaría de ser gratuito.

Ahora podemos visitar la aplicación en nuestro navegador con `heroku open`.

```
$ heroku open
```

**Nota:** ¡Verás una pagina de error! Hablaremos de eso en un minuto.

Esto abrirá una url como <https://djangogirlsblog.herokuapp.com/> en tu navegador, y al momento verás probablemente una página de error.

El error que ves es porque cuando desplegamos en Heroku creamos una nueva base de datos que está vacía. Necesitamos ejecutar los comandos `migrate` y `createsuperuser`, tal como hicimos en PythonAnywhere. Esta vez, se ejecutan vía una línea de comandos especial en nuestro ordenador, `heroku run`:

```
$ heroku run python manage.py migrate
```

```
$ heroku run python manage.py createsuperuser
```

Este comando te pedirá que elijas un nombre de usuario y password otra vez. Éstas serán tus credenciales en la página de administración de tu sitio web.

Refresca tu navegador, y ¡ahí está! Ahora ya sabes cómo desplegar en dos plataformas de hosting diferentes. Elige tu favorita

## Cómo alojar un proyecto Android

## Cómo publicar tu app

Importante: A partir de agosto de 2021, las apps nuevas deberán publicarse con Android App Bundle en Google Play. Ahora, las apps nuevas de más de 150 MB son compatibles con la entrega de funciones en Play o Play Asset Delivery.

La publicación es el proceso general por el cual tus apps de Android se ponen a disposición de los usuarios. Cuando publicas una aplicación para Android, realizas dos tareas principales:

### Preparas la app para su lanzamiento.

Durante el paso de preparación, compilas una versión de lanzamiento de tu aplicación que los usuarios pueden descargar e instalar en sus dispositivos Android.

### Realizas el lanzamiento de la app para los usuarios.

Durante el paso de lanzamiento, publicitas, vendes y distribuyes la versión de actualización de tu aplicación para los usuarios.

En esta página, se proporciona una descripción general del proceso que debes seguir a medida que te preparas para publicar tu app. Si tienes pensado publicarla en Google Play, también debes leer la [lista de tareas de lanzamiento de Google Play](#).

## Cómo preparar tu app para el lanzamiento

La preparación de tu aplicación para el lanzamiento es un proceso de varios pasos que implica las siguientes tareas:

### Configurar tu app para el lanzamiento

Como mínimo, debes quitar las llamadas de Log y el atributo android:debuggable de tu archivo de manifiesto. También debes proporcionar valores para los atributos android:versionCode y android:versionName, que se encuentran en el elemento <manifest>. Asimismo, es posible que debas configurar varios ajustes para cumplir con los requisitos de Google Play o adaptar cualquier método que uses para lanzar tu aplicación.

Si usas los archivos de compilación de Gradle, puedes emplear el tipo de compilación de lanzamiento a fin de configurar tus ajustes de compilación para la versión publicada de tu app.

- Compilar y firmar una versión de actualización de tu app



- Puedes usar los archivos de compilación de Gradle con el tipo de compilación release a fin de compilar y firmar una versión de actualización de la app. Consulta la sección Cómo compilar y ejecutar desde Android Studio.
- 
- Probar la versión de lanzamiento de tu app
- Antes de distribuir tu app, debes probar por completo la versión de actualización en al menos un teléfono celular y una tablet de destino.
- 
- Actualizar los recursos de la app para el lanzamiento
- Asegúrate de que todos los recursos de la app, como los archivos y gráficos multimedia, estén actualizados y se incluyan con tu app o por etapas en los servidores de producción correspondientes.
- 
- Preparar servidores y servicios remotos de los cuales dependa tu app
- Si tu app depende de servidores o servicios externos, debes controlar que sean seguros y estén listos para la producción.

Es posible que debas realizar varias tareas como parte del proceso de preparación. Por ejemplo, deberás obtener una clave privada para firmar la aplicación, crear un ícono y también es conveniente que prepares un contrato de licencia para el usuario final (CLUF) a fin de proteger a tu persona, organización y propiedad intelectual.

Cuando termines de preparar tu aplicación para el lanzamiento, contarás con un archivo .apk firmado que puedes distribuir a los usuarios.

## Cómo lanzar tu app a los usuarios

Puedes lanzar tus apps de Android de diferentes maneras. Por lo general, las apps se lanzan a través de un mercado como Google Play, pero también puedes hacerlo en tu propio sitio web o enviarlas de forma directa a los usuarios.

## Cómo realizar el lanzamiento a través de un mercado de apps

Si deseas distribuir tus apps al público más amplio posible, recurrir a un mercado de apps como Google Play es la opción ideal.

Google Play es el principal mercado de apps para Android y la mejor forma de distribuir tus aplicaciones a un público mundial amplio. Sin embargo,

puedes distribuir tus apps en cualquier mercado que desees o incluso usar varios.

### **Cómo lanzar tus apps en Google Play**

Google Play es una plataforma de publicación sólida que te permite publicitar, vender y distribuir tus aplicaciones para Android a usuarios de todo el mundo. Cuando lanzas tus aplicaciones a través de Google Play, tienes acceso a un paquete de herramientas de desarrollo que te permiten analizar tus ventas, identificar las tendencias de mercado y controlar los destinatarios a los cuales se distribuyen tus aplicaciones. También tienes acceso a varias funciones que aumentan las ganancias, como la facturación integrada y las licencias para aplicaciones. La amplia matriz de herramientas y funciones, junto con varias características de la comunidad de usuarios finales, convierten a Google Play en el principal mercado de compra y venta de aplicaciones para Android.

Lanzar tu aplicación en Google Play es un proceso simple que consta de tres pasos básicos:

### **Preparar materiales de promoción**

Para aprovechar al máximo las capacidades de mercado y publicidad de Google Play, debes crear materiales de promoción para tu aplicación, como capturas de pantalla, videos, gráficos y texto promocional.

### **Configurar opciones y subir recursos**

Google Play te permite orientar tu app hacia un grupo de usuarios y dispositivos de todo el mundo. Cuando configuras varios ajustes de Google Play, puedes elegir los países a los que desees llegar, los idiomas que quieras usar y el precio que pretendes imponer en cada región. También puedes configurar detalles de la ficha, como el tipo de aplicación, la categoría y la clasificación del contenido. Cuando terminas de configurar las opciones, puedes subir tu material de promoción y tu aplicación como borrador (sin publicar).

### **Publicar la versión de lanzamiento de tu app**

Si sabes que los ajustes de la publicación están configurados correctamente y que la aplicación subida está lista para lanzarse al público, solo haz clic en la opción Publicar de Play Console, y estará disponible para la descarga en todo el mundo en algunos minutos.

### **Cómo realizar un lanzamiento a través de un sitio web**

Si no quieres lanzar tu app en un mercado como Google Play, puedes dejarla disponible para la descarga en tu propio sitio web o servidor, incluso en un servidor privado o empresarial. Para ello, primero prepara tu aplicación para el lanzamiento de la manera normal. Luego, aloja el archivo APK listo para el lanzamiento en tu sitio web y ofrece un vínculo de descarga a los usuarios.

Cuando naveguen hasta el vínculo de descarga desde sus dispositivos Android, se descargará el archivo, y el sistema Android comenzará a instalarlo en el dispositivo automáticamente. Sin embargo, el proceso de instalación comenzará automáticamente solo si los usuarios tienen su dispositivo configurado para permitir que se instalen apps desde fuentes desconocidas.

Aunque lanzar tu app en tu propio sitio web es relativamente sencillo, puede resultar ineficiente. Por ejemplo, si deseas monetizar tu aplicación, tendrás que procesar y seguir todas las transacciones financieras tú mismo, y no podrás usar el servicio de Facturación integrada de Google Play para vender productos integrados. Además, no podrás usar el servicio de licencias para evitar la instalación y el uso no autorizado de tu aplicación.

### **Aceptación de los usuarios para la instalación de fuentes y apps desconocidas**

Android protege a los usuarios de descargas e instalaciones inadvertidas de apps que provengan de ubicaciones que no sean tu tienda de aplicaciones, como Google Play (que es confiable). Hasta que el usuario habilita la instalación de apps de otras fuentes, Android las bloquea. El proceso de aceptación depende de la versión de Android que se ejecute en el dispositivo del usuario:

Captura de pantalla que muestra la configuración para aceptar la instalación de apps desconocidas de otras fuentes.

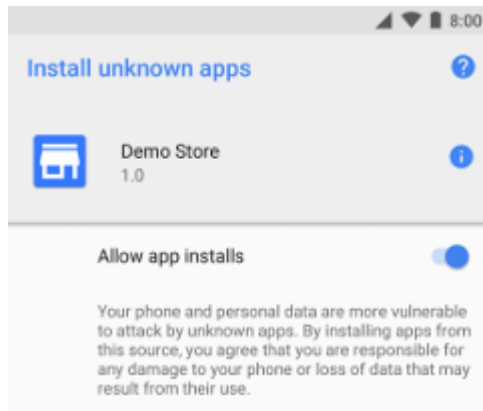


Figura 1: La pantalla de configuración del sistema Instalar apps desconocidas, en donde los usuarios otorgan permiso a una fuente específica para la instalación de apps desconocidas.

En los dispositivos que ejecutan Android 8.0 (nivel de API 26) y versiones posteriores, los usuarios deben navegar a la pantalla de configuración del sistema Instalar apps desconocidas para habilitar la instalación de apps de una fuente específica.

En los dispositivos que ejecutan Android 7.1.1 (nivel de API 25) y versiones anteriores, los usuarios deben habilitar la configuración del sistema Fuentes desconocidas o permitir la instalación única de una app desconocida.

### Instalar apps desconocidas

En los dispositivos que ejecutan Android 8.0 (nivel de API 26) y versiones posteriores, los usuarios deben otorgar permiso para instalar apps de una fuente que no sea una tienda de apps propia. Para ello, deben habilitar la opción de configuración Permitir la instalación de apps para esa fuente dentro de la pantalla Instalar apps desconocidas del sistema. La Figura 1 ilustra este proceso.

**Nota:** Los usuarios pueden cambiar esta configuración para una fuente específica en cualquier momento. Por lo tanto, una fuente que instala apps desconocidas siempre debe llamar a `canRequestPackageInstalls()` a fin de verificar si el usuario le otorgó permiso a esa fuente para que instale apps desconocidas. Si este método muestra el valor `false`, la fuente debe solicitarle al usuario que vuelva a habilitar la configuración Permitir la instalación de apps para esa fuente.

### Fuentes desconocidas

En la captura de pantalla, se muestra la configuración para aceptar la descarga y la instalación de apps desde fuentes desconocidas.

Figura 2: La opción de configuración Fuentes desconocidas, que determina si los usuarios pueden instalar apps que no se descargan desde Google Play

En los dispositivos que ejecutan Android 7.1.1 (nivel de API 25) y versiones anteriores, los usuarios deben habilitar la opción de configuración Fuentes desconocidas en Configuración > Seguridad, como se muestra en la figura 2, a fin de permitir la instalación de apps de fuentes externas.

Nota: Cuando los usuarios intentan instalar una app desconocida en un dispositivo que ejecuta Android 7.1.1 (nivel de API 25) o versiones anteriores, algunas veces el sistema le pregunta al usuario si solo desea permitir la instalación de una app en particular. En la mayoría de los casos, si la opción está disponible, los usuarios deben permitir la instalación de una sola app desconocida a la vez

En cualquier caso, los usuarios deben hacer este cambio de configuración antes de poder descargar e instalar apps desconocidas en sus dispositivos.

Nota: Algunos proveedores de red no permiten a los usuarios instalar apps de fuentes desconocidas.

## ¿Cómo confeccionar un tutorial técnico en youtube para demostrar tus capacidades?

**Paso 1: Determina el tema a tratar para el tutorial. Elige el tema**

**Paso 2: Investiga el tema aunque lo conozcas.**

Cuando hacemos un tutorial, sin dar una última revisada a nuestro tema, se nos pueden ir cosas que no hemos considerado, tales como alguna actualización, nuevas herramientas que existan o algo que no tengamos presente y que puede ser información útil al momento de que nuestra audiencia vea el tutorial.

**Paso 3: Anticipa las herramientas que vas a necesitar y consíguelas.**

**Paso 4: Prepara la edición y grabación.**

Recuerda: Sé tú mismo.

**Paso 5: Graba o filma el tutorial.**

**Paso 6: Exporta el video tutorial.**

**Paso 7: Titular el video.**

## ¿Cómo presentar tu proyecto en youtube?

Para que un video sea efectivo, en un mundo lleno de contenido, debe ser **práctico, útil y relevante**, para eso debes centrarte en aquellos aspectos más relevantes de tu proyecto y contar el proceso destacando lo más interesante y dar énfasis en aquellos puntos de aprendizaje, contenido que puede ser útil para los demás. Debes tener esto en cuenta a la hora de construir el guión del video.

## Referencias:

<https://kinsta.com/es/base-de-conocimiento/que-es-github/>

<https://www.behance.net/gallery/89953611/UX-Design-Portfolio-2020>

[https://www.top10.com/hosting/comparison?utm\\_source=google&kw=servicio%20de%20hosting%20gratis&c=417678790445&t=search&p=&m=e&adpos=&dev=c&devmod=&mobval=0&network=g&campaignid=146480388&adgroupid=6679005708&targetid=kwd-302497782842&interest=&physical=1003302&feedid=&a=2805&ts=hi&topic=&test=&gclid=Cj0KCQiAuvOPBhDXARIsAKzLQ8G1Mm5Dt6T3rDMbNJg4-8gm7w5FY0PYPOKX1mBKV8DgDNI0twEDOuwaAoXFEALw\\_wcB](https://www.top10.com/hosting/comparison?utm_source=google&kw=servicio%20de%20hosting%20gratis&c=417678790445&t=search&p=&m=e&adpos=&dev=c&devmod=&mobval=0&network=g&campaignid=146480388&adgroupid=6679005708&targetid=kwd-302497782842&interest=&physical=1003302&feedid=&a=2805&ts=hi&topic=&test=&gclid=Cj0KCQiAuvOPBhDXARIsAKzLQ8G1Mm5Dt6T3rDMbNJg4-8gm7w5FY0PYPOKX1mBKV8DgDNI0twEDOuwaAoXFEALw_wcB)

<https://jcalderon.files.wordpress.com/2008/02/app-jsf.pdf>

<https://argentinaenpython.com/django-girls/extensiones-tutorial/heroku/>

<https://enmihomestudio.com/como-hacer-video-tutoriales-para-youtube-en-pc-guia-paso-a-paso/>

<https://parzibyte.me/blog/2019/12/11/publicar-app-node-js-heroku/>

<https://developer.android.com/studio/publish?hl=es-419>