



# *Universidad Nacional de Loja*

## Análisis y Diseño de Sistemas

Heroku — Cloud Application Platform. Despliegue de aplicaciones a partir del GitHub

Antonio Aguilar ECINF7224 Manuel Armijos ECINF7225  
Ricardo Jumbo ECINF7207 Pablo Sarango ECINF7226  
Luís Solano AFINF7205 Jefferson Vera ECINF7434.

Ingeniería en Sistemas  
pgsarangou@unl.edu.ec

# Introducción

## Tema

### Introducción

### Agenda

### Objetivos

### Desarrollo

### Conclusiones

### Bibliografía

### Licencia

A continuación se mostrará el despliegue de una aplicación sencilla a partir de GitHub. Para el despliegue se ha usado Heroku. Heroku ofrece un servicio Paas (Plataform as a Service) en donde actualmente se pueden desplegar aplicaciones desarrolladas en Ruby, Node.js, Java, Python, Clojure y Scala. En Heroku definen una aplicación como el código fuente de la aplicación junto a sus dependencias, indicadas, por ejemplo, en el Gemfile de Ruby o el pom.xml de Maven. Inicialmente, a cada aplicación se le asigna un dyno, que es como llaman en Heroku a sus servidores ligeros donde ejecutan el comando indicado por el usuario. El uso de dynos permite una gran escalabilidad a nuestra aplicación, ya que en cualquier momento podemos contratar más o menos dynos según las necesidades que tengamos. [1].

# Agenda

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

- Heroku
  - Servicios en Heroku
  - Importancia.
  - Ventajas.
  - Desventajas.
  - Arquitectura.
  - Registro
  - Heroku Toolbelt
- Despliegue de aplicación
  - GitHub
  - Requisitos
  - Despliegue en Heroku
- GitHub y Heroku
  - Sincronización
  - Commit en GitHub y Heroku.
  - Despliegue automático.
  - Despliegue manual.

# Objetivos

## Tema

Introducción  
Agenda  
**Objetivos**  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

- Desplegar una aplicación a partir de GitHub en Heroku.
- Demostrar el uso de plataformas para el desarrollo colaborativo.
- Examinar el empleo Plataformas como Servicio o PaaS.

# Heroku. Servicios

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

Actualmente Heroku ofrece los siguiente servicios.



Node.js



Ruby



Java



PHP



Python



Go



Scala



























Clojure

# Heroku. Servicios

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

## Base de datos.

  <b>Gráfico Historia</b> Enterprise Neo4j Gráfico bases de datos como servicio	  <b>Redis Nube</b> De clase empresarial Redis para desarrolladores	  <b>AT &amp; T M2X</b> Servicio de AT & T de las series de tiempo de almacenamiento de...	  <b>ClearDB MySQL</b> La base de datos de alta velocidad para aplicaciones alimentadas...
  <b>Instaclustr</b> Bases de datos Apache Cassandra NoSQL alojados y...	  <b>RedisGreen</b> Redis servidores calidad de producción con el apoyo y análisis...	  <b>openredis</b> Fiable Hosting Redis.	  <b>GrapheneDB</b> Neo4j Gráfico de base de datos como un servicio
  <b>Tesoro de datos</b> Big Data Analytics Hadoop en Heroku	  <b>Heroku Postgres</b> Base de datos confiable y de gran alcance como un servicio basad...	  <b>Bucketeer</b> Una forma sencilla de utilizar Amazon S3 desde su aplicació... <span>Beta</span>	  <b>Redis To Go</b> #1 Proveedor Redis con más de 50.000 casos Redis.

# Heroku. Servicios

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## Almacenes de datos.

   <b>PostgreSQL Estudio</b> Basado en la Web UI para las instancias Heroku Postgres	  <b>RedisMonitor</b> Seguimiento de la salud de su servidor Redis y prevenir los...	  <b>Adminium</b> Completa interfaz de administración hecha y derecha...	  <b>Heroku Conectar</b> Servicio Salesforce sincronización de datos
   <b>Vaquero</b> Wrangler es un cliente de SQL y el informe de intercambio de...	  <b>Xplenty</b> Servicio de Integración de Datos. Simplificado.	  <b>Autobus</b> Servicio de copia de seguridad automática para PostgreSQL y...	

# Heroku. Servicios

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## EMAIL/SMS.

Incorporar y hacer un seguimiento de mensajería segura y fiable a los usuarios o dentro de su propia aplicación.



### Mailtrap por Railware

Pruebas de correo electrónico seguro para la puesta en escena...



### SparkPost

Entrega de correo electrónico y análisis construidas para los...



### Mail2Webhook

Recibe el correo entrante como HTTP POST para su aplicación...



### Blower.io SMS

Fácil de mensajes SMS



### SendGrid

Correo Entrega. Simplificado.



### CloudMailin

Escalable Correo electrónico entrante para su Heroku App



### Mailgun

El motor de automatización de correo electrónico



### Mandrill por MailChimp

Entrega de correo electrónico escalable y análisis para las...



# Heroku. Servicios

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

Servicios para ayudarle en pruebas de rendimiento y la calidad de sus aplicaciones.



BlazeMeter

Plataforma de pruebas de rendimiento



loader.io

Pruebas de carga basadas en la nube simple para sus aplicaciones...



Selva QA

Prueba increíblemente simple.



Bombardero aéreo

Pruebas de carga de la Nube

# Heroku. Servicios

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## Seguridad.



### Tinfoil Seguridad

Sitio web de análisis de seguridad que es simple, asequible y...



### Acelerado SSL

Certificado SSL Compra e instalación como un servicio.



### SSL

El cifrado SSL para https:// urls.



### Impresión segura

Rotación Impresión segura y automática agraciada

# Heroku. Importancia

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

La importancia se presenta a la hora de desplegar. Nos brinda una multitud de opciones para elegir desde hostings compartidos a servidores privados, o incluso la famosa nube. Todo dependerá del alcance y necesidades de nuestra aplicación y sobre todo, del presupuesto que tengamos.

Es una plataforma que nos permite trabajar con aplicaciones web desarrolladas en ruby on rails (y otras tecnologías), también nos permite que concentremos esfuerzos en la creación de aplicaciones y olvidemos de la configuración del servidor.

# Heroku. Importancia

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

Hay que tener en cuenta que un “dynos” es un proceso que atiende a peticiones para las aplicaciones; entre mas dynos hay, mas peticiones atendidas habrá. En cuanto al tiempo, es óptimo para un proyecto inicial, Heroku podría ayudar a acelerar el tiempo de comercialización del producto en etapas posteriores.

En cuanto a la prueba de aplicación en Heroku no es obligatorio alcanzar un mínimo en el número de pruebas realizadas. Los nuevos proyectos que duran más de dos meses son desarrollados usando pruebas desde cero.

# Heroku. Ventajas

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

- La mayor ventaja de este paradigma es la rapidez con la que se puede publicar una aplicación a la nube.
- Con un comando “git push heroku master”, la aplicación está lista para recibir peticiones. No hay que invertir tiempo en configurar servidores, firewalls, ni bases de datos.
- Todo esto viene con un coste adicional asociado, pero generalmente para aplicaciones o equipos pequeños realmente vale la pena ya que ahorra muchos problemas y dolores de cabeza que pueden presentarse si además hay que mantener la infraestructura.

# Heroku. Desventajas

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

- La mayor desventaja con servicios como Heroku es la falta de personalización y optimizaciones que pueden realizarse cuando hay acceso más abierto a la infraestructura.
- Para una aplicación pequeña esto no es un problema representativo, pero para una aplicación con decenas de millones de visitas al día, las optimizaciones a nivel de infraestructura pueden representar la diferencia entre funcionar o colapsar, además de representar grandes ahorros de coste.
- Entre algunas de las alternativas a Heroku encontramos a Google App Engine, Openshift, AppFog y DotCloud.

# Heroku. Arquitectura

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

## WebSocket



## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## WebSocket

- El protocolo WebSocket es una tecnología que forma la base de aplicaciones web modernas en tiempo real.
- Proporciona un canal bidireccional para la entrega de datos entre clientes y servidores.
- Da la flexibilidad de una conexión TCP con el modelo de seguridad adicional y metadatos incorporados en el protocolo HTTP.



## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## Tiempo de respuesta HTTP

Cuando una conexión dyno se haya establecido

- Las solicitudes HTTP tienen una ventaja de 30 segundos iniciales para que el proceso web devuelva datos de respuesta (ya sea la respuesta completado o una cierta cantidad de datos que eindicque que el proceso esta activo).
- Los procesos que no envían los datos de respuesta dentro de la ventaja inicial de 30 segundos verán un error.

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## El modelo de Proceso

- Es una abstracción sencilla y potente para ejecutar programas de servidor.
- Aplicado a las aplicaciones web, el modelo de proceso nos da una manera única de pensar en dividir nuestras cargas de trabajo y la ampliación en el tiempo.
- El modelo de procesos se utiliza para los dynos web, trabajadores y todos los otros tipos de dinamómetros.

## Tema

Introducción

Agenda

Objetivos

**Desarrollo**

Conclusiones

Bibliografía

Licencia

## Tipos de procesos vs dinámómetros

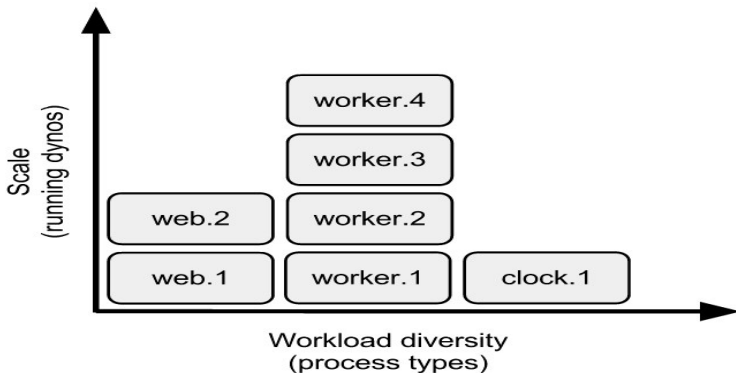
- Es un tipo de proceso que realiza prototipo de la que se instancian uno o más dinámómetros.
- Esto es similar a la forma en que una clase es el prototipo de la cual se instancian uno o más objetos en la programación orientada a objetos.

# Heroku. Arquitectura

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## Tipos de procesos vs dinamómetros



# Heroku. Arquitectura

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## Slug

- Son comprimidos y copias preenvasadas de su aplicación optimizados para la distribución al dyno manager.
- Cuando git push a Heroku, su código es recibido por el compilador babosa que transforma su repositorio en una babosa. Escalar una aplicación a continuación descargas y amplía la babosa a un banco de pruebas para su ejecución.

## Dynos

El compilador Slug es invocado por un git pre-recepción de gancho, que sigue estos pasos:

- Crear una copia nueva de la cabeza de la rama principal.
- Quite cualquier cosa especificada en un archivo `.slugignore` de nivel superior.
- Descarga, construir e instalar las dependencias locales como se especifica en el archivo de generación (por ejemplo, Gemfile, package.json, requirements.txt, pom.xml, etc.) con la herramienta de gestión de la dependencia con el apoyo del idioma (por ejemplo Bundler, npm, pip, Maven).
- Empaquete el archivo slug final.

### Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## Tamaño Slug

- El tamaño máximo permitido slug (después de la compresión) es de 300 MB.

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## Dynos

- Un banco de pruebas(Dynos) es un contenedor ligero Linux que ejecuta un único comando especificado por el usuario.
- Un banco de pruebas puede ejecutar cualquier comando disponible en su entorno por defecto (lo que la oferta el Cedar stack) o en su aplicación slug (una copia comprimida y pre-envasados de su aplicacion y sus dependencias)



# Heroku. Arquitectura

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## Tipos de Dynos

Heroku corre dinámómetros de tres maneras diferentes:

- Dynos Web
- Dynos Trabajadores
- Dynos on-off

# Heroku. Arquitectura

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## Dynos Web

- Los Dynos Web son dinamómetros de la "web" su tipo de proceso se define en su Procfile. Solamente los dinamómetros web reciben tráfico HTTP desde los routers de Heroku.

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## Dynos Trabajadores

- Puede hacer cualquier tipo de proceso que este declarado en su Procfile, distinto de "web".
- Los Dynos trabajadores se utilizan normalmente para trabajos en segundo plano, sistemas de colas y los trabajos programados.
- Puede tener varios tipos de Dynos trabajadores en la aplicación. Por ejemplo, uno para los trabajos urgentes y otra para trabajos de larga duración.

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## Dynos One-Off

- Los Dynos Ono-Off son dinamómetros temporales que se pueden ejecutar unifamiliares, o con su entrada / salida conectado a su terminal local.
- Están cargados con su último lanzamiento. Pueden ser utilizados para manejar las tareas administrativas, como las migraciones de bases de datos y sesiones de consola.
- También pueden ser utilizadas para ejecutar ocasionalmente trabajo de fondo, con el Programador de Heroku.

# Heroku. Arquitectura

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## Tipos de Dynos

Dyno Type	Sleeps	Professional Features	Memory (RAM)	CPU Share	Dedicated	Compute (I)
free	minimum 6hrs / day	no	512MB	1x	no	1x-4x
hobby	no	no	512MB	1x	no	1x-4x
standard-1x	no	yes	512MB	1x	no	1x-4x
standard-2x	no	yes	1024MB	2x	no	4x-8x
performance-m	no	yes	2.5GB	100%	yes	12x
performance-l	no	yes	14GB	100%	yes	50x

# Heroku. Arquitectura

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

## Tamaño de los Dynos

- Heroku acredita automáticamente cada aplicación con 750 dyno-hora gratuitas al mes, que están claramente identificados en su factura.
- Esta asignación se puede utilizar para cualquier tipo de banco de pruebas de cualquier tamaño del banco de pruebas.
- Tenga en cuenta que 2X dinamómetros consumen el doble de Dyno-horas libres comparado con las 750 dyno-hora 1X dynos, 2x se ejecutará de forma gratuita durante 375 horas.
- PX consumen 16 veces más Dyno-horas libres por horas como 1X dinamómetros.

# Heroku. Arquitectura

## Tamaño de Dynos

Dyno Size	Memory (RAM)	CPU Share	Multitenant	Compute (I)	Price/dyno-hour
1X	512MB	1x	yes	1x-4x	\$0.05
2X	1024MB	2x	yes	4x-8x	\$0.10
PX	6GB	100%	no	40x	\$0.80

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

## Dyno Sleeping y App de Recharging

- Las Aplicaciones que utilizan el tipo de banco de pruebas libres tienen un sleeping(sueño) y un comportamiento de recarga.
- Los Dinamómetros gratis dormiran cuando un banco de pruebas no recibe ningún tráfico web por un período de tiempo.
- Si un banco de pruebas libre supera una cuota de 18 horas de actividad durante una ventana de 24 horas, se verá obligada a recargar.



## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## Dyno Sleeping y App de Recharging

Una aplicación está activa si alguna parte de la aplicación se está ejecutando. Por ejemplo, si algunos de las siguientes condiciones son verdaderas:

- Hay un banco de pruebas web que está recibiendo tráfico.
- Hay un banco de pruebas trabajador que se está ejecutando.
- Un banco de pruebas on-off esta corriendo. Por ejemplo, se inició a través de la CLI o el planificador.

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

## Sleeping

- Si una aplicación tiene un Dyno web y no recibe ningún tráfico en un período de 30 minutos, el Dyno web va a dormir.
- Además de dormir el dyno web, el Dyno trabajador (si está presente) también va a dormir.
- Si un Dyno web recibe tráfico, se activará de nuevo después de un breve retraso.
- Si la aplicación tiene Dyno trabajador que fue a dormir antes de ampliarse, será ampliado de nuevo también.

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

## Recharging

- Las aplicaciones en el estado recarga" se cierran durante 6 horas.
- Si un Dyno (ya sea web, trabajador o one-off) se ejecutarán durante este período de tiempo. Cualquier tráfico web recibida en este estado devolverá una página de error.

# Heroku. Registro

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

Para el registro en Heroku solo debemos rellenar el siguiente formulario.

Sign up for free and experience Heroku today

**Free account**  
Create apps, connect databases and add-on services, and collaborate on your apps, for free.

**Your app platform**  
A platform for apps, with app management & instant scaling, for development and production.

**Deploy now**  
Go from code to running app in minutes. Deploy, scale, and deliver your app to the world.

First name

Last name

Email

Company name

**Create Free Account**

Signing up signifies that you have read and agree to the [Terms of Service and Privacy Policy](#).

Already have an account? [Log In](#)

# Heroku. Heroku Toolbelt

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

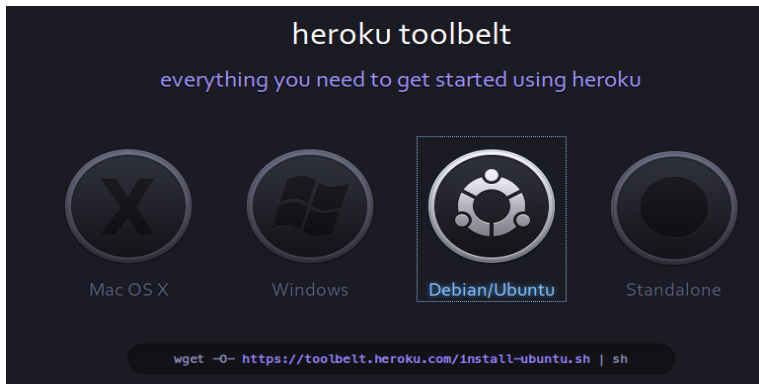
Es una herramienta de línea de comandos para trabajar con la plataforma Heroku en OS X, Windows y Debian/Ubuntu. Para la instalación debemos dirigirnos al siguiente enlace [LINK](#). Seleccionar nuestro sistema operativo y seguir el procedimiento correspondiente.

# Heroku. Heroku Toolbelt

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

En este caso usaremos Ubuntu. Por lo que ejecutaremos en consola el comando de la imagen.

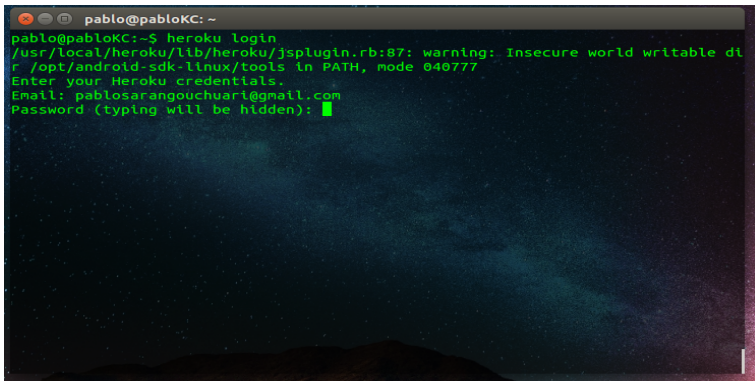


# Heroku. Heroku Toolbelt

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

Una vez instalado debemos logearnos. Para ello en una terminal escribimos *heroku login*. Nos pedirá el email con el que nos registramos y nuestra contraseña. Una vez hecho esto ya podemos desplegar nuestra aplicación en Heroku.

A terminal window with a dark background and a starry space theme. The prompt is 'pablo@pabloKC: ~'. The command 'heroku login' has been entered. The output shows a warning about an insecure world writable directory, followed by prompts for email and password. The email 'pablosarangouchuari@gmail.com' has been entered. The password prompt is visible with a redacted character.

```
pablo@pabloKC: ~  
pablo@pabloKC:~$ heroku login  
/usr/local/heroku/lib/heroku/jsplugin.rb:87: warning: Insecure world writable di  
r /opt/android-sdk-linux/tools in PATH, mode 040777  
Enter your Heroku credentials.  
Email: pablosarangouchuari@gmail.com  
Password (typing will be hidden): █
```

# Despliegue. GitHub

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

A partir de un proyecto existente en GitHub haremos el despliegue en Heroku.

The screenshot shows the GitHub interface for the repository 'pablosarango / ejercicio'. At the top, there are buttons for 'Watch' (1), 'Star' (0), and 'Fork' (0). Below this, a summary bar indicates '7 commits', '1 branch', '0 releases', and '1 contributor'. The main content area shows the 'master' branch selected, with a list of files: 'web' (Heroku Auto, an hour ago), 'Procfile' (Subida2, 9 hours ago), 'README.md' (primer commit, 11 hours ago), and 'composer.json' (Subida, 9 hours ago). Below the file list is the 'README.md' content, which displays the word 'ejercicio' in a large, bold font. On the right side, there are links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Pulse', and 'Graphs'. At the bottom right, the 'HTTPS clone URL' is provided as 'https://github.cc', with a note that it can also be cloned with 'HTTPS' or 'Subversion'. A 'Download ZIP' button is also present.

pablosarango / ejercicio

7 commits 1 branch 0 releases 1 contributor

Branch: master ejercicio / +

pablosarango Heroku Auto Latest commit 45b2c97 an hour ago

web	Heroku Auto	an hour ago
Procfile	Subida2	9 hours ago
README.md	primer commit	11 hours ago
composer.json	Subida	9 hours ago

README.md

## ejercicio

<> Code

Issues 0

Pull requests 0

Pulse

Graphs

HTTPS clone URL

https://github.cc

You can clone with HTTPS or Subversion.

Download ZIP



# Despliegue. Requisitos

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

Para este ejemplo usaremos una aplicación sencilla en PHP. Para que Heroku reconozca que nuestra aplicación está en PHP debemos definir en la raíz de nuestro proyecto un archivo llamado *composer.json*. Una opción es instalar *Composer*, el cual es un gestor de dependencias para PHP. Dada la sencillez de nuestro aplicativo no usaremos dependencias por lo que no es necesario la instalación del gestor. No obstante si es necesario la existencia del archivo *composer.json*.

Dependiendo del lenguaje de nuestra aplicación deberemos seguir unos u otros pasos para que Heroku sea capaz de ejecutarla. Esta información la podemos encontrar en la página oficial de Heroku.

# Despliegue. Requisitos

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

La estructura de nuestro archivo *composer.json* es la siguiente.

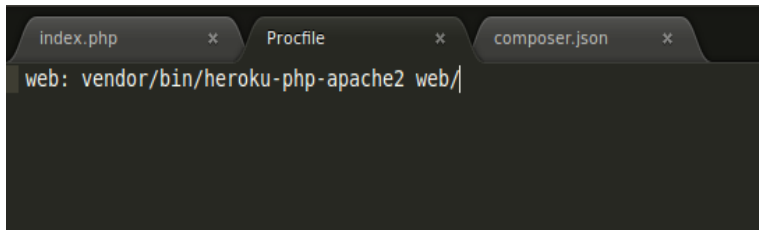
```
{
  "name": "Ejercicio integración entre GITHUB y HEROKU",
  "author":
  {
    "name": "Pablo Sarango",
    "email": "pgsarangou@unl.edu.ec",
    "homepage": "http://terminalec.com"
  }
}
```

# Despliegue. Requisitos

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

Además del archivo *composer.json* es necesario crear un archivo llamado *Procfile* el cual nos permitirá tener acceso a nuestra aplicación en Heroku. Este archivo no tiene extensión. El contenido de este archivo variará según el tipo de aplicación que estemos corriendo. Tener en cuenta que nosotros tenemos nuestro código php en una carpeta llamada "web". Razón por la cual especificamos esto en nuestro archivo Procfile



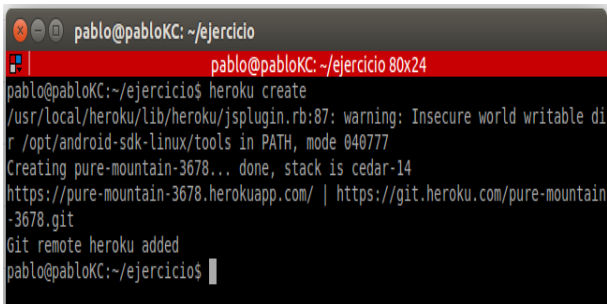
```
index.php x Procfile x composer.json x
web: vendor/bin/heroku-php-apache2 web/
```

# Despliegue. Heroku

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

En una terminal y posicionados en el directorio de nuestro proyecto ejecutamos el siguiente comando *heroku create*. Ahora ya tenemos la dirección url de nuestro proyecto así como un repositorio remoto de nuestra aplicación en Heroku. Heroku establecerá un nombre aleatorio a nuestro proyecto.

A terminal window titled 'pablo@pabloKC: ~/ejercicio' with a red title bar. The terminal shows the command 'heroku create' being executed. The output includes a warning about an insecure world writable directory, the creation of a new app named 'pure-mountain-3678' on the 'cedar-14' stack, and the generation of two URLs: 'https://pure-mountain-3678.herokuapp.com/' and 'https://git.heroku.com/pure-mountain-3678.git'. It also shows 'Git remote heroku added' and the prompt returning to 'pablo@pabloKC:~/ejercicio\$'.

```
pablo@pabloKC: ~/ejercicio
pablo@pabloKC:~/ejercicio$ heroku create
/usr/local/heroku/lib/heroku/jsplugin.rb:87: warning: Insecure world writable di
r /opt/android-sdk-linux/tools in PATH, mode 040777
Creating pure-mountain-3678... done, stack is cedar-14
https://pure-mountain-3678.herokuapp.com/ | https://git.heroku.com/pure-mountain
-3678.git
Git remote heroku added
pablo@pabloKC:~/ejercicio$
```

# Despliegue. Heroku

Ahora procedemos a subir el código fuente de nuestra aplicación con el comando *git push heroku master*.

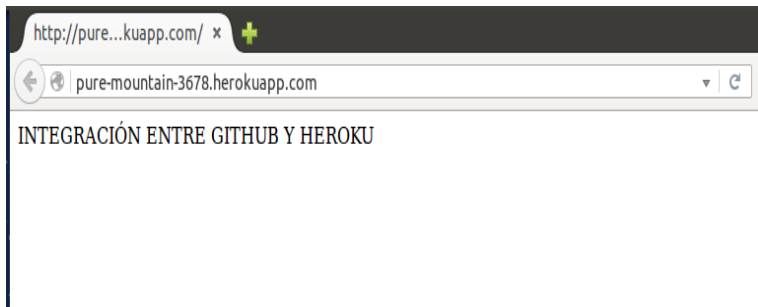
```
pablo@pabloKC: ~/ejercicio
pablo@pabloKC:~/ejercicio$ git push heroku master
Counting objects: 10, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 1009 bytes | 0 bytes/s, done.
Total 10 (delta 0), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> PHP app detected
remote: -----> No runtime required in 'composer.json', defaulting to PHP 5.6.15.
remote: -----> Installing system packages...
remote: -----> - PHP 5.6.15
remote: -----> - Apache 2.4.16
remote: -----> - Nginx 1.8.0
remote: -----> Installing PHP extensions...
remote: -----> - zend-opcache (automatic; bundled)
remote: -----> Installing dependencies...
remote: -----> Composer version 1.0.0-alpha11 2015-11-14 16:21:07
remote: -----> Loading composer repositories with package information
remote: -----> Installing dependencies
remote: -----> Nothing to install or update
remote: -----> Generating optimized autoload files
remote: -----> Preparing runtime environment...
remote: -----> NOTICE: No Procfile, using 'web: vendor/bin/heroku-php-apache2'.
remote: -----> Discovering process types
remote: -----> Procfile declares types => web
remote: -----> Compressing... done, 72.8MB
remote: -----> Launching... done, v3
remote: -----> https://pure-mountain-3678.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy.... done.
To https://git.heroku.com/pure-mountain-3678.git
 * [new branch]      master -> master
pablo@pabloKC:~/ejercicio$
```

# Despliegue. Heroku

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

Ahora nuestra aplicación ya se encuentra corriendo en Heroku. Para acceder a ella podemos ingresar a la url que vimos anteriormente o podemos ejecutar en consola el comando *heroku open*.



# GitHub y Heroku. Sincronización

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

Lo siguiente es entrar a nuestra cuenta de Heroku y seleccionar nuestra aplicación. Para la sincronización debemos irnos a la sección *Deploy* y acceder al tab de *GitHub*. Posteriormente debemos buscar nuestro repositorio y conectarlo.



Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

Search for a repository to connect to

pablosarango

ejercicio

Search

Missing an organization? [Ensure Heroku Dashboard has organization access](#).

 pablosarango/ejercicio

Connect



App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to  [pablosarango/ejercicio](#)

Disconnect

✓ Releases in the [activity feed](#) link to GitHub to view commit diffs

# GitHub y Heroku. Commit

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

Cuando hayamos realizado cambios a nuestro código y queremos que estos se vean reflejados en nuestra aplicación en Heroku debemos hacer *commit* a GitHub y actualizar en Heroku. Para ello escribimos los comandos de la imagen. Se nos pedirá nuestro usuario de GitHub y contraseña.

```
pablo@pabloKC: ~/ejercicio
pablo@pabloKC: ~/ejercicio 80x24
pablo@pabloKC:~/ejercicio$ git add .
pablo@pabloKC:~/ejercicio$ git commit -m "heroku"
En la rama master
Su rama está delante de «origin/master» para 2 commits.
(use "git push" to publish your local commits)

nothing to commit, working directory clean
pablo@pabloKC:~/ejercicio$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

In Git 2.0, Git will default to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.
```



# GitHub y Heroku. Despliegue Automático

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

Para reflejar los cambios de manera automática en Heroku nos dirigimos a nuestra aplicación, después a la sección *Deploy* y al tab *GitHub*. Ahí tendremos que activar el despliegue automático.

### Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

### Enable automatic deploys from GitHub

Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically**; be sure that this branch is always in a deployable state and any tests have passed before you push. [Learn more](#).

☐ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Enable Automatic Deploys

# GitHub y Heroku. Despliegue Automático

## Tema

Introducción  
Agenda  
Objetivos  
**Desarrollo**  
Conclusiones  
Bibliografía  
Licencia

Con esto cada vez que hagamos *push* nuestra aplicación se actualizará en Heroku automáticamente.

### Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

☒ Automatic deploys from `master` are enabled

Every push to `master` will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch in GitHub is always in a deployable state and any tests have passed before you push. [Learn more.](#)

☐ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Disable Automatic Deploys

# GitHub y Heroku. Despliegue Manual

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

En nuestra aplicación nos vamos a la sección *Deploy* y al tab de *GitHub*. Al final de la página encontraremos una sección que la que haciendo click en *Deploy Branch* se reflejarán los cambios que hemos hecho a nuestra aplicación.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

 master

Deploy Branch

Receive code from GitHub



Build master [Show build log](#)



Deploy to Heroku



Your app was successfully deployed.

 View

# GitHub y Heroku. Commit

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
Licencia

Ahora ya estarán disponibles los cambios en nuestra aplicación.



# Conclusiones

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
**Conclusiones**  
Bibliografía  
Licencia

El empleo de una Plataforma como Servicio, en este caso Heroku, permite a los desarrolladores abandonar el uso de servidores para sus aplicaciones. Este tipo de servicio les ofrece un gran abanico posibilidades. La mayoría posibilitan usarlos de manera gratuita con aplicaciones pequeñas, además, brinda la oportunidad de pagar cuando la aplicación esté terminada. Dependiendo de la escalabilidad de la aplicación estos servicios proporcionan la capacidad de comprar nuevas características y en algunos casos pagar solo por el tiempo que se use más requerimientos.

# Conclusiones

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
**Conclusiones**  
Bibliografía  
Licencia

Permite enfocarse en el desarrollo de la aplicación al no tener que preocuparse por la configuración de servidores y la implementación de estos. Otra ventaja es que el despliegue se hace a través de Git. Por lo que si trabajamos con Git nos resultará fácil adaptarnos al funcionamiento de Heroku. Heroku también detecta automáticamente qué tipo de aplicación estamos subiendo, así, si detecta que es una aplicación en Ruby on Rails lanzará rails server, o python app/manage.py si estamos usando Django.

# Bibliografía

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
**Bibliografía**  
Licencia



F. J. Lopez-Pellicer, R. Béjar, M. A. Latre, J. Nogueras-Iso, and F. J. Zarazaga-Soria, “GitHub como herramienta docente.” Universitat Oberta La Salle, July 2015, pp. 66–73. [Online]. Available: <http://upcommons.upc.edu/handle/2117/76761>

# Licencia

## Tema

Introducción  
Agenda  
Objetivos  
Desarrollo  
Conclusiones  
Bibliografía  
**Licencia**





Muchas Gracias

