

# A Jr. Developer's Guide to Software Engineering December 2018

- Ahora: Jr. Developer, Pythonista/Djangonaut
- Mi pasado: Accountant & uni instructor w/ an MBA

## Who Else Am I?

- DjangoCon US Organizer
- DEFNA North American Ambassador
- @DEFNADO for more info

# ¿Aprender a Escribir Código?

## ¿Por qué?

## ¿Cómo?

## Primer Problema

- Career changers seek internships or "Jr. Developer" jobs
- Jr. Developer == ???
- “¿Cuando estoy preparado?”

# Primer Problema

- ¿Cómo me convierto en periodista? Más vocabulario? No.
- Más vocabulario de Python != mejor ingeniero de software

## Segundo Problema

- How do you mentor a new developer?

# Dos Estilos para Escribir Código

## Programación

- Solo
- Autor
- Menos robusto

## Ingeniería de Software

- Los grupos
- Otra gente
- Más robusto

# Interjección(!): programadores “de verdad”

- If your code runs & does what you want it to do, you’re a “real” coder.
- Qué estilo necesitas?



# Temas de Discusión

- Sistema de Control de Versiones
- Documentación
- Pruebas de Software
- Gestión de la Dependencia/Despliegue
- Entorno de Desarrollo de Software

# Sistema de Control de Versiones: ¿Por qué?

- Protect yourself from mistakes
- Enable yourself to try new things
- Learn to work on a team

# Sistema de Control de Versiones: ¿Cuando?

- More code than you want to retype
- Working code you're scared to break

# Sistema de Control de Versiones: ¿Cómo?

- Utilízalo solo
- Bifurcar y reintegrar
- Git != Github

# Sistema de Control de Versiones: ¿Cómo?

- Pro Git Chapters 1-3
- <https://git-scm.com/book/en/v2>
- Free!
- Ochenta por ciento Git todos los días

# Documentación: ¿Por qué?

- Refine your thinking
- Help your future self
- Help your teammates help you

# (!) Código de Auto-Documentación

- Es un poco verdad...pero, no.
- Hacer buenos nombres es difícil
- Start to think about it now

## Documentation: ¿Cuando?

- Writing code using a new technique
- Writing functions, especially if they call other functions



# Documentation: ¿Cómo?

- Docstrings for functions (minimums)
  - ¿Qué hace la función?
  - ¿Qué necesita la función? (con tipo de datos)
  - ¿Qué envía la función? (con tipo de datos)

¿Por qué Pruebas?

Muchas.

Muchas.

Razones.

# ¿Por qué Pruebas?: El Panorama

- Tests -> Pruebas Automatizadas
- Pruebas Automatizadas -> CI
- CI -> CD (Continuous Delivery)
- CD -> CD (Continuous Deployment)

# ¿Por qué Pruebas? Para Nuevos Programadores

- Protection from future changes breaking things
- Assurance past fixes still work
- Develop good habits for teamwork

## Pruebas: ¿Cuándo?

- As soon as possible
- Arte y ciencia: empieza a practicar ahora

# Pruebas: ¿Cómo?

- doctest & unittest: builtin
- pytest: *pip install pytest*

# Gestión de la Dependencia y Despliegue

- Building more complex software
- Building software for others

# Gestión de la Dependencia y Despliegue

- Virtual Env
- Containers
- Virtual Machines (VMs)



# Otro

- Entorno de Desarrollo de Software
  - CLI vs. GUI: can you navigate?
  - Terminal multiplexer
    - screen, tmux, byobu
  - A shell language: Bash or Powershell

# Temas de Discusión (una vez más)

- Sistema de Control de Versiones
- Documentación
- Pruebas de Software
- Gestión de la Dependencia/Despliegue
- Entorno de Desarrollo de Software

# Finding New Developers

To:

- Engineering Managers
- Dev Leads
- Others trying to hire developers...

# Questions/Comments?

- @Transition on Twitter
- [kojoidrissa.com](http://kojoidrissa.com)
- [https://github.com/kojoidrissa/  
pydaymx\\_2018](https://github.com/kojoidrissa/pydaymx_2018)