

Víctor Pascual

vikpascual.es

github.com/vikpascual

linkedin.com/in/victor-pascual-muñoz-48177111a

Valencia
(+34) 644 412 988
vikpascual@gmail.com

EXPERIENCIA

Instituto de Ingeniería Energética (IIE) de la Universitat Politècnica de Valencia

DICIEMBRE DEL 2022 - DICIEMBRE DEL 2024 (2 AÑOS)

- Programación y Control de distintas fuentes renovables (solar, eólica, etc..) para su integración en la red.
 - Predicción de la demanda eléctrica
 - Pandas, NumPy, SkLearn, Plotly, Dash
- Web Scraping con Python
- FastAPI
- Flask Web
- NodeJS, EJS, Bootstrap, HTML, CSS, JavaScript
- Electron JS
- Reverse proxy ExpressJS
- Administración, gestión y diseño de Bases de Datos
- Administración con Bash Scripting/Powershell y Python
- Copias de seguridad

Odec Centro de Cálculo y Aplicaciones Informáticas SA, Valencia : *Administrador de sistemas, Formación Dual*

SEPTIEMBRE DEL 2019 - MAYO DEL 2020 (9 MESES)

- Administración de Vmware Virtualization Technology (VMware ESXi)
- Desarrollo páginas web (Liferay, Wordpress, HTML, JavaScript, CSS, PHP, SQL)
- Administración de bases de datos
- Windows Server
- Administración con Bash Scripting/Powershell y Python
- Administración de redes (DHCP, VLAN, ARP, DNS)
- Monitorización en red de equipos (SNMP, SpiceWorks, The Dude, WMI)
- Copias de seguridad

FORMACIÓN

Universidad Politécnica de Valencia (Etsinf), Valencia: *Grado en Ingeniería Informática rama ciencias de la computación*

SEPTIEMBRE DEL 2020 - ACTUALMENTE

APTITUDES

Python, FastAPI, React, NodeJS, ElectronJS, SQL, HTML, CSS, JavaScript, PHP, SQL, Bootstrap, Tailwind CSS

Web Scraping

Arduino

Scripting

Autodidacta

Trabajo en equipo

IDIOMAS

Castellano (Nativo)

Catalán (Nativo)

Inglés (Intermedio-Bajo)

IES ABASTOS, Valencia: Administrador de Sistemas Informáticos en RED (ASIR), Modalidad Dual

SEPTIEMBRE DEL 2018 - JUNIO DEL 2020

Curso realizado en modalidad dual, es decir, media jornada en la empresa y la otra media jornada en clase durante 6 meses

PROYECTOS

Inasolar (<https://inasolar.es/>)

Aplicación para gestionar la instalación de distintas fuentes renovables, así como predecir la demanda eléctrica y actuar en consecuencia.

- Python, Pandas, Numpy, Plotly
- Dash, Flask
- NodeJS, ExpressJS, EJS, ElectronJS
- Bootstrap
- FastAPI
- SQL
- Consultas a API, Scripting

Admin Oasis (<https://github.com/vikpascual/Admin-Oasis>)

Aplicación web con el objetivo de centralizar todas las tareas de administración posibles en un único sitio.

- PHP
- HTML, CSS Y JavaScript
- JQuery
- AJAX
- MySQL
- WMI, SNMP, SCTP, LDAP, Active Directory
- Apache

Aproximación de funciones mediante grafos (top secret)

Aproximador de funciones mediante generación de grafos. La generación puede ser aleatoria, utilizando un árbol de búsqueda monte carlo(MCTS) o mediante algoritmos genéticos. El objetivo es que varios clientes generen grafos hasta que alguno encuentre la solución.

- React
- Tailwind
- Go
- MongoDB

Portfolio (<https://vikpascual.es/>)

Portfolio donde muestro lo que he estado haciendo estos años. Está hecho para ser lo más rápido posible haciendo uso de las mejores prácticas (Diseño responsive, traducciones i18n, tema claro-oscuro, lighthouse, SEO, etc...)

- Astro JS
- TypeScript
- Tailwind CSS
- Nginx

- Responsive
- i18n (traducciones)
- Tema claro y oscuro
- SEO

Localizador láser de cuerpos celestes

(<https://github.com/vikpascual/Laser-planet-locator>)

Proyecto desarrollado en **Python** y **Arduino**. Se consulta la **API Horizons** de la NASA para obtener el ángulo de elevación en el cielo y la dirección respecto a las coordenadas magnéticas del cuerpo celeste elegido. Una vez obtenida esta información, se envía a Arduino mediante una conexión serial y mueve los motores en consecuencia.

- Arduino
- Python
- Consultas API

Q-Learning PONG (IA)

(<https://github.com/vikpascual/Q-learning-Pong>,
<https://www.youtube.com/watch?v=HKUVfGaPp1s>)

El juego de Pong implementado con 2 agentes controlados por inteligencia artificial en **Python**. Ambos agentes trabajan juntos de manera cooperativa para evitar que la pelota pase de largo. Utilizamos el algoritmo **Q-Learning** para entrenar a los agentes en este juego. Para la comunicación de los agentes se ha utilizado una arquitectura cliente-servidor mediante **sockets TCP**.

- Sockets TCP
- Python
- PyGame

Simulador de vuelo espacial

(https://github.com/vikpascual/nave_espacial,
<https://www.youtube.com/watch?v=s9MXc5UuTJo>)

Simulador de vuelo utilizando **C++** y **OpenGL**. Física, iluminación y navegación realizadas mediante operaciones con matrices vectoriales.

- C++
- OpenGL
- Matemáticas