



José Fabián
Soltero Escobar

Programador Full Stack

Contacto

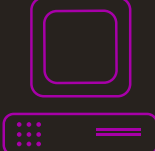
Número : 744 452 0905
Correo: fabian_94_3@hotmail.com

Lenguajes, frameworks
y tecnologías que uso:

- >_ Python
- >_ Django
- >_ JavaScript
- >_ React
- >_ React Native
- >_ TailwindCSS
- >_ PostgresDB
- >_ MongoDB
- >_ Golang
- >_ Docker
- >_ MATLAB

Idioma :

- >_ Ingles intermedio



Habilidades:

- >_ Desarrollo de microservicios con Python y Node.js
- >_ Conocimiento en la nube de AWS, Lambda, ApiGateway, RDS, CloudWatch, EC2
- >_ Conocimiento en el desarrollo mobile usando React Native
- >_ Desarrollo Web y APIRest con Django
- >_ Conocimiento Base de datos con Postgres y MongoDB
- >_ Manejo de Control de Versiones de código con Git y Github
- >_ Conocimiento en contenedores Docker
- >_ Uso de Metodologías Agiles
- >_ Conocimiento en la suite de Atlassian: Jira, Confluence

Estudios:

Instituto Politécnico Nacional – UPIITA
2015 - 2019 (Trunca)

Instituto Politécnico Nacional – ESIME
2014 - 2015

Preparatoria Americana de Acapulco
2010 - 2013

Experiencia profesional:

Mis Métricas

Fecha: Enero 2022 – Marzo 2023
Puesto: Desarrollador Full Stack

Freelancer

Fecha: 2021 – 2022
Sistema para resolver ecuaciones lineales usando Python
Generador de Punto de venta usando Django y TailwindCSS

Sitio web y redes sociales:

- <https://www.github.com/fabian-ss>
- <https://randomcodeweb.netlify.app/>
- <https://www.youtube.com/c/RandomMusicD>
- <https://www.linkedin.com/in/fabian-soltero-escobar>

Experiencia en Web

Mi sitio Web:	Por parte del IPN-UPIITA, tome un curso de programación web enfocado en HTML, CSS y JavaScript. El proyecto final de ese curso que fue mi página web, la cual termine pasando a React.
URL	https://randomcodeweb.netlify.app/
Página Ciudad Paleta Costa Azul	Página diseñada para un negocio de paletas de hielo. Este proyecto esta hecho con TaiwindCSS y React, desplegado usando github pages, el diseño es responsivo
URL	https://fabian-ss.github.io/Ciudad-Paleta-Costa-Azul-Web/

Experiencia como Full Stack

Lista de reproducción en Youtube con demos de Apps en React Native:

- <https://youtube.com/playlist?list=PLPyV1h7t5kar2d1Bsw9TJ8Es4Ag9Exqe8>

Pago de Servicios:	El objetivo del proyecto fue, que, en la aplicación de Métricas, se pudieran hacer pago de servicios, como pagar la luz, agua, gobierno, o hacer recargas telefónicas, entre otros pagos.
Desarrollador: Full Stack	Mi participación en el proyecto fue como desarrollador Full Stack, trabajé el frontend usando React Native, y el backend usando microservicios en la nube de AWS
	El frontend lo hice siguiendo las especificaciones del diseñador UX/UI, con el que estuve en constante comunicación para seguir su diseño, y mejorar la experiencia de usuario
	Tecnologías: <ul style="list-style-type: none">- JavaScript (frontend)- TailWindCSS (frontend)- Node (backend)- Python (backend)- Lambda - ApiGateway - Cloudwatch - Git- Versionado de AWS - Capas AWS - RDS
Punto de venta:	El objetivo del proyecto fue crear un backend y un frontend en Django para administrar las ventas, llevar un control de inventario y ver estadísticas del mejor producto dependiendo de la fecha que se elija
Desarrollador: Full Stack	En un servidor local cree una aplicación Django conectado a Postgres como base de datos, expuse la dirección ip y el puerto de la app para que de forma local cada dispositivo conectado a la red tuviera acceso al punto de venta
Herramientas: <ul style="list-style-type: none">- Python- Lambda- ApiGateway- Cloudwatch- Git- Versionado AWS- Capas AWS- RDS- SOAP	La interfaz fue diseñada usando TailwindCSS, donde aparecía el producto, el costo y el total del carrito
	Desarrollo del proyecto: <ul style="list-style-type: none">>_ Establecí las definiciones>_ En un diagrama de flujo establecí el flujo global de cada proceso, como el CRUD de los productos, el CRUD de las ventas, el carrito de compra en la interfaz>_ Usando Postgres creé una el esquema y las tablas del proyecto>_ Usando Visual Studio Code cree la aplicación de Django>_ Versione cada cambio usando Git>_ Hice el frontend usando HTML, TailwindCSS y JavaScript

Experiencia Cloud en AWS

Lambda:	Sé implementar scripts en Python o Node.js, implementar o usar las capas, y, conectarme a bases de datos usando RDS
Api Gateway:	Sé asignar enpoints a Lambdas, para después exponerlas mediante su implementación
RDS:	Puedo crear bases de datos relacionales como Postgres
CloudWatch:	Puedo observar el comportamiento que tuvo una Lambda durante su ejecución, de esta forma puedo comprobar que el microservicio funciona correctamente, o incluso detectar un error o excepción durante la ejecución

Métologia para trabajar los microservicios o el backend

Desarrollo del proyecto:

- >_ Establezco las definiciones para cada microservicio
- >_ Usando RDS con Postgres defino y creo el esquema y tabla donde se guardará la información
- >_ Crear un diagrama de flujo en Lucidchart para definir el flujo global de cada microservicio
- >_ En Visual Studio Code, creo un entorno local de trabajo donde creo y desarrollo el script de cada microservicio, ya sea de Python o Node, incluye la conexión a la DB
- >_ Una vez funcione el microservicio a nivel local lo cargo a Lambda con sus respectivas capas
- >_ Uso Git para ir versionando cada script en local, y el versionado nativo de Lambda
- >_ En Api Gateway expongo los endpoints con cada uno de sus métodos y parámetros necesarios
- >_ Creo la documentación para detallar como se consume el servicio

Experiencia en Backend

Microservicios:	Creación y mantenimiento de microservicios para una Fintech. Microservicios exclusivos para la App movil de la fintech. Microservicios para exponer a terceros. Estos son algunos de los microservicios que desarrolle:
Desarrollador: Backend	<ul style="list-style-type: none">- Transferencia de dinero entre tarjetas de la fintech (Backend)- Transferencia de dinero, de la tarjeta de la fintech a bancos externos (Backend)- Realizar pagos con cargo a la tarjeta (Backend)- Realizar reembolsos a la tarjeta (Backend)- Obtener los movimientos por tarjeta (Backend)- Consumo de servicios SOAP- Servicio escrito en Python, que genera un token del tipo JWT, con cifrado mediante llaves RSA (Backend)
	Tecnologías: <ul style="list-style-type: none">- Python - Lambda - ApiGateway - Cloudwatch - Git- Versionado AWS - Capas AWS - RDS - SOAP
Adelanto de pago:	El objetivo del proyecto fue dar un adelanto de pago a los usuarios que lo solicitaran, y que fueran candidatos.
Desarrollador: Backend	Mi participación fue como desarrollador de microservicios, tuve que evaluar si los usuarios que solicitaban el adelanto eran candidatos, y de ser así, calcular el porcentaje que se les podía otorgar
	Herramientas: <ul style="list-style-type: none">- Python - Lambda - ApiGateway - Cloudwatch - Git- Versionado de AWS - Capas AWS - RDS