

Me complace presentarles este PDF que contiene una mirada profunda hacia algunas de mis experiencias, habilidades y enfoque en el mundo del desarrollo.

Este PDF tiene un enfoque de **Entrevista Exclusiva** y es para que descubran algunos Detalles de Mi Trayectoria.

Comencemos la entrevista...

1. Si eres un candidato viable, cuál es tu disponibilidad para ingresar a trabajar?

Estoy disponible de inmediato para comenzar a trabajar y no tengo ningún compromiso personal o profesional que afecte mi disponibilidad.

2.Cuál es tu expectativa salarial?

Mi expectativa salarial es no menos de \$10,00 USD/hora. Sin embargo, estoy dispuesto a negociar en función de los detalles específicos del puesto y los beneficios adicionales que se ofrezcan.

3. Cuántos años tienes utilizando Mern Stack y cómo has utilizado estas tecnologías en proyectos anteriores?

He trabajado con las tecnologías individuales que lo componen en diferentes proyectos.

Durante los últimos 7 años, he tenido la oportunidad de trabajar con cada tecnología individual que conforma el stack MERN.

Estas incluyen **MongoDB (1 año)**, **React (5 años)**, **Node.js (3 años)** y **Express.js (3 años)**.

MongoDB:

Trabaje con MongoDB en un proyecto de gestión de Hospitales, médicos y usuarios.

Esta aplicación la desarrolle usando Angular, Node.js y Express.js como marco principal para crear un CRUD completo; implemente Mongoose Schema Model junto a

referencias entre colecciones para asignación personalizada; validaciones usando MongoDB; búsquedas comunes (findByIdAndUpdate/Delete) y avanzadas (proyección campos + búsqueda simultánea + búsquedas con expediciones regulares); recursividad mediante populate(); contador documental countDocuments(); ordenamiento sort(), skip() and limit() para paginación resultados finalmente conexión al backend a través del servicio Atlas Cloud Service.

URL del proyecto: <https://dennysjmarquez.medium.com/angular-10-mean-google-auth-jwt-lazyload-upload-de-archivos-guards-pipes-zona-admin-bfa2e5ef9074#fdd2>

React:

Mi experiencia en React como parte del stack MERN incluye la creación de aplicaciones web y móviles utilizando React, React Native y Next.js.

He trabajado como React Developer en Epidata durante 6 meses, como React SSR - Next.js Developer en Credit Agile, Inc durante 2 años y 4 meses, y como React Native Developer durante 2 años.

Me he enfocado en diseñar e implementar características únicas para cada proyecto involucrado.

Algunos de mis conocimientos usados son: React Function Component, class Component, ciclos de vida, React Hooks: (useState, useEffect, useCallback, useReducer, useContext, useRef, useLayoutEffect, useMemo, React.memo), Portal, createElement, isValidElement, HOC, flushSync.

También, he redactado artículos sobre buenas prácticas en React, como "React la guía máxima de buenas prácticas jamás concebida, Principios SOLID, El acrónimo STUPID, Clean Code y Code Smell" y "La Arquitectura Hexagonal: un enfoque para desarrollar aplicaciones robustas en React, Angular y Vue con ejemplos de código

1. React la guía máxima de buenas prácticas jamás concebida, Principios SOLID, El acrónimo STUPID, Clean Code y Code Smell
<https://dev.to/dennysjmarquez/react-la-guia-maxima-de-buenas-practicas-jamas-concebida-principios-solid-el-acronimo-stupid-clean-code-y-code-smell-5317>

2. La Arquitectura Hexagonal: un enfoque para desarrollar aplicaciones robustas en React, Angular y Vue con ejemplos de código
<https://dev.to/dennysjmarquez/react-la-guia-maxima-de-buenas-practicas-jamas-concebida-principios-solid-el-acronimo-stupid-clean-code-y-code-smell-53l7>

Node.js y Express.js

Mi experiencia con Node.js y Express.js como parte del stack MERN se enfoca o especializa en implementar APIs RESTful.

He trabajado con Node y Express en React SSR - Next.js Developer en Credit Agile, Inc durante 2 años y 4 meses.

Como parte de mi experiencia con Node y Express, desarrollé una API interna en Next.js, la cual usé como una API Gateway para ocultar el servidor que me proporcionaba la data al igual que el token. Utilicé sesiones para manejar los datos de forma segura entre request, middlewares personalizados para extraer la información de dichas sesiones y exponerla a Next.js, Memorystore para almacenar las sesiones en memoria del servidor e implementación de express-session, express-fileupload, next-i18next/middleware, express.json(), node-cache, CRUD completo haciendo uso de next-connect y de Segmentos Dinámico [...param].js, y Segmentos de Captura Total [...param]].js y CORS entre otros recursos necesarios.

También he trabajado con Node y Express en los proyectos como:

1. Node.js y Express.js | Sessions, ataques CSRF como hacerlo y evitarlo, Token CSRF e implementación
https://www.youtube.com/playlist?list=PLIVwhETeDf2Ge8oPBp19d_9OYcRf5KwM
Código fuente: <https://github.com/dennysjmarquez/express-manager-session-security-csrf-token>
2. Un sencillo servidor para el proyecto de (Angular 13 | Plataforma de Movies)
Node.js, Express.js, cors, este servidor sirve un archivo JSON como dB en un único

endPoint protegido por un token de auth0., Db de Películas con su Título, descripción, tipo de película, imagen y fecha

Código fuente: <https://github.com/dennysjmarquez/angularStreamingServer>

3. Back-End para el proyecto de (Angular 10 | angular-adv-adminpro), Node.js, Express.js, MongoDB, Mongoosejs, CRUD, JWT, express-validator, google sign in, Paginaciones de datos, y mucho más...

<https://dennysjmarquez.medium.com/angular-10-mean-google-auth-jwt-lazyload-upload-de-archivos-guards-pipes-zona-admin-bfa2e5ef9074#fdd2>

Código fuente: <https://github.com/dennysjmarquez/angular-adv-adminpro-backend>

Y sumo este artículos: Todo lo que necesitas saber sobre API Rest: Glosario de términos esenciales y más.

<https://dev.to/dennysjmarquez/todo-lo-que-necesitas-saber-sobre-api-rest-glosario-de-terminos-esenciales-y-mas-29pc>

En este artículo hago referencia a las API orientada a HTTP API Rest.

Aunque no he utilizado estas tecnologías juntas en un solo proyecto, tengo una comprensión general de cómo funcionan juntas y cómo se pueden aplicar.

4. ¿Cómo has integrado APIs de terceros en tus proyectos anteriores y qué consideraciones tuviste en cuenta al hacerlo?

En mis proyectos he integrado APIs de terceros considerando la seguridad como una prioridad ante todo.

Normalmente, las APIs de terceros son un sistema seguro y fiable para implementar en los proyectos, gracias a que ya han sido desarrolladas y probadas; sin embargo, nunca tendremos control total sobre ellas. Esto deja un margen para posibles problemas de seguridad que debemos considerar al integrarlas.

En mis proyectos en los que he integrado APIs de terceros, siempre he tenido la precaución de almacenar las credenciales o API Key en archivos .env y de implementarlas dentro de un try catch. Es bueno ver el historial de seguridad de la API

y los fallos de seguridad que ha tenido. Esto nos da una idea de qué tan segura puede llegar a ser implementarla, a la hora de elegir una API de terceros. Si la API no es vieja y es un proyecto que se ha mantenido a lo largo del tiempo, todo eso hay que considerarlo. También hay que considerar el tiempo de reacción de la API y la carga que puede recibir, ya que esto puede afectar la aplicación que se está desarrollando.

Una API que no está diseñada para manejar una carga alta puede ser vulnerable a ataques de denegación de servicio (DoS) o ataques de inundación que podrían interrumpir sus servicios, al igual que una API que tarda demasiado en responder puede ser vulnerable a ataques de tiempo de espera, en los que un atacante podría tratar de explotar la lentitud de la API para realizar ataques de fuerza bruta o inyección de código malicioso.

Recientemente, adquirí conocimientos de **CyberSecurity - Ethical Hacking**, **NIST**, controles de seguridad y **OWASP TOP 10** en [Seguridad Cero Academy](#), y puedo hacer un uso mejor de la seguridad en mis APIs Rest.

OWASP ofrece una guía para la implementación correcta de APIs de terceros en:

1. <https://owasp.org/API-Security/editions/2023/en/0x11-t10/>
2. <https://owasp.org/API-Security/editions/2023/en/0xaa-unsafe-consumption-of-apis/>

Además, también he aprendido la importancia de tener en cuenta la **deuda tecnológica** en proyectos que se alargan en el tiempo. Si no se actualizan regularmente las APIs o SDK utilizados, podríamos terminar con APIs desactualizadas y SDKs, lo que representa un grave problema de seguridad tanto para nuestras aplicaciones como para nuestras APIs. Por lo tanto, es crucial mantenerse al día con los últimos avances tecnológicos y actualizar regularmente los componentes clave del proyecto para garantizar la seguridad y eficiencia a largo plazo.

Les dejo un artículo que redacté sobre la deuda tecnológica en proyectos que se alargan en el tiempo:

<https://dev.to/dennysjmarquez/la-deuda-tecnologica-en-nuestros-proyectos-13p>

Hay muchas cosas que aprender y mejorar.

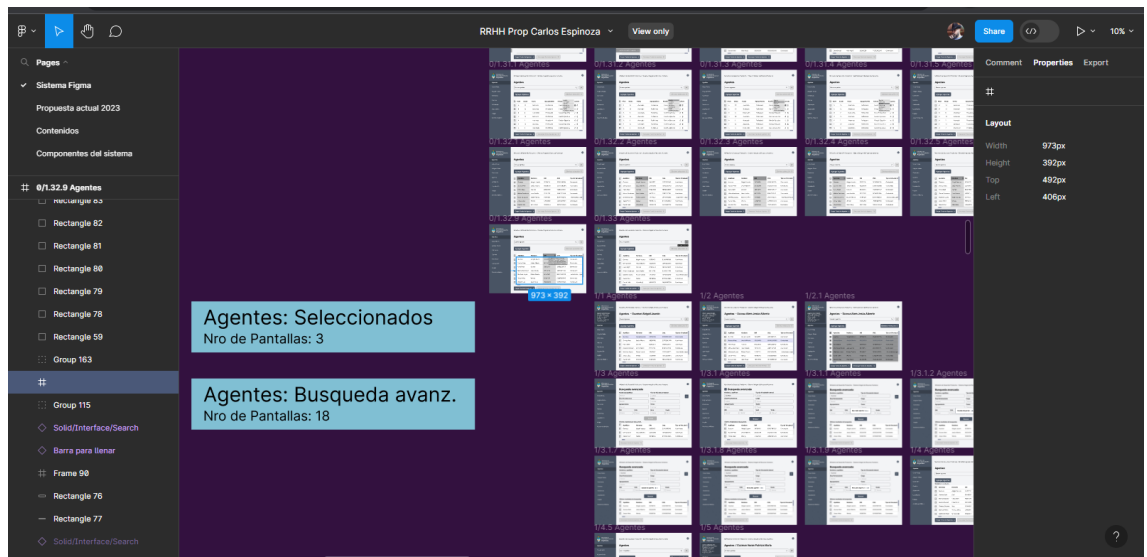
5. ¿Has integrado API's de exchange de cryptomonedas (Binance, Bitso, etc)

No, y aunque no he tenido experiencia directa integrando estas APIS, estoy interesado en aprender más sobre este tema y estoy dispuesto a adquirir las habilidades y conocimientos necesarios para hacerlo en el futuro.

6. ¿Cuánto tiempo tienes utilizando Tailwind, qué desafíos has enfrentado al trabajar con el y cómo los has superado?

He estado utilizando Tailwind CSS durante los últimos 6 meses. Durante mi último empleo en **Epidata**, tuve la oportunidad de aplicar esta herramienta en la replicación de una maqueta de Figma en React, donde también implementé la librería [@headlessui/react](#).

Aunque he disfrutado trabajando con Tailwind CSS, también he enfrentado algunos desafíos, claro que sí, y en algunos casos me tomó más tiempo encontrar la clase adecuada para el estilo que quería aplicar, pero superé esto consultando la documentación de Tailwind CSS, misma admito es muy completa.



7. ¿Puedes hablar sobre tu experiencia en el diseño y la implementación de soluciones de almacenamiento de datos, así como en el manejo de bases de datos relacionales y no relacionales como MongoDB y Firebase?

Mi experiencia trabajando con Firestore, Firebase Realtime Database, Cloud Functions y MySQL ha sido muy interesante.

1. Trabajé en una aplicación React Native en **Credit Agile Inc**, donde implementé Firebase Realtime Database y Cloud Functions.

En este desarrollo decidí usar la base de datos en tiempo real de Google, ya que la aplicación móvil que desarrollé tendría un sistema de Trackit de seguimiento, y una lista de notificaciones.

El sistema de Trackit utilizaba la geolocalización para mostrar su ubicación en tiempo real en un mapa de Google datos necesarios y útil para la compañía recuperadora.

Implementé un sistema de notificaciones push mediante **Cloud Functions**. Cuando se crea una nueva entrada en la tabla de notificaciones, la Cloud Functions realiza dos acciones. Primero, realiza un mantenimiento automático de la base de datos, eliminando notificaciones antiguas de más de un año. Segundo, genera y envía una notificación push a la aplicación asociada a la notificación. De esta manera, el usuario recibe la notificación en la aplicación y también puede verla en una lista de notificaciones que se actualiza automáticamente gracias a la base de datos en tiempo real.

Para proteger la base de datos, utilice las reglas que permiten que solo el usuario administrador (auth.uid) pueda escribir y leer los datos.

Implementade una solución útil y eficiente para la gestión de préstamos y pagos de Credit Agile Inc.

2. He utilizado Firestore, he desarrollado una base de datos para gestionar mi sistema de publicaciones, mi portafolio y mi perfil profesional en mi sitio web <https://dennysjmarquez.dev/>

Durante este proceso, he trabajado con técnicas de paginación offset y limit, select para traer solamente la información necesaria, uso de where en general, consultas compuestas y arreglos para validar o filtrar según una lista de valores y criterios de búsqueda, creación de índices, uso de orderBy y count() para obtener el número de documentos en una colección.

Además, he utilizado un headless CMS utilizando el SDK de Flamelink para gestionar mis publicaciones y contenido, y he implementado reglas de Firestore para asegurar la protección de la base de datos, verificando si el usuario que lee o escribe en la base de datos es el administrador.

Para optimizar el sitio web y el uso de la base de datos, creé un sistema de caché que funcionaba tanto en el backend como en el frontend.

Este sistema permite mostrar el sitio web casi de forma inmediata, la cache no dependía de sesiones, ya que se almacena en la memoria del servidor por un máximo de 30 minutos.

Si alguien realizaba una consulta que ya está en caché, el servidor devuelve los datos almacenados en caché en lugar de consultar la base de datos.

Aplicando así técnicas como paginación y limitación de resultados para evitar problemas de rendimiento.

Gracias a la implementación de la paginación de Firestore, logré resolver problemas de lentitud, ya que antes traía todos los resultados y con esto se limitaba a un máximo de 10 resultados por página, y como mi sitio web estaba alojado en Netlify, solucioné el error 501 que ocurría cuando la aplicación tardaba mucho en cargar.

3. En mi proyecto final del **Diplomado de Programación Web en Uneweb Instituto**, desarrollé un **CMS similar a WordPress** utilizando **MySQL** como gestor de base de datos.

Para simplificar la estructura de la base de datos y mejorar la eficiencia en la gestión de categorías y subcategorías, implementé la técnica de Nested Sets. Esta técnica me permitió almacenar todas las categorías y subcategorías en una sola tabla, estableciendo las columnas left y right para la relación jerárquica entre ellas.

Creé una tabla de relaciones utilizando la taxonomía para gestionar la información de diferentes taxonomías en una sola tabla.

Utilizando las tablas "relation" y "data", sigo una estructura que permite establecer relaciones entre objetos y datos utilizando taxonomías. Esta estructura puede brindar flexibilidad y facilidad de gestión al organizar y clasificar la información en el sistema.

Esto permite tener una estructura de base de datos más simple y evitar la creación de múltiples tablas separadas.

También utilicé dos tablas para los post, una para los propios post y otra para agregar los postmeta asociados al mismo, lo que me permitió crear múltiples campos adicionales y asociarlos a un post.

Para la conexión y gestión de la base de datos, utilicé la biblioteca Doctrine DBAL, incluyendo la clase Doctrine\DBAL\DriverManager. Para maximizar la seguridad en la gestión de consultas a la base de datos, implementé técnicas como el método prepare y bindParam para evitar ataques de inyección de SQL.

Proyecto: <https://dennysjmarquez.medium.com/un-cms-parecido-a-wordpress-y-lo-llame-scorpion-cms-abd9d7005cde>

Código fuente: <https://github.com/dennysjmarquez/scorpioncms>

8. ¿Cómo has realizado pruebas unitarias en tus proyectos anteriores y cuál es tu enfoque para garantizar la eficiencia y la capacidad de respuesta del software?

1. He trabajado con las librerías **jest** y **superTest** para implementar pruebas unitarias y de integración en proyectos backend basados en Node.js, como este:

<https://github.com/dennysjmarquez/tdd-en-nodejs-guia-de-tests-con-jest>

En este proyecto incluí tanto pruebas unitarias como de integración.

2. También he aplicado pruebas unitarias y de integración en este proyecto

<https://dennysjmarquez.medium.com/angular-10-mean-google-auth-jwt-lazyload-upload-de-archivos-guards-pipes-zona-admin-bfa2e5ef9074#db81>

3. Donde use Angular 10 y **Jasmine 3.10**. Además, realicé este curso **¡Introducción al Testing de Software para Principiantes!**

<https://www.udemy.com/course/introduccion-al-testing-de-software-para-principiantes/>

4. Donde aprendí que existen metodologías para probar el software, las aplicaciones y más y esto me ayudó a entender e implementar las estrategias de prueba adecuadas.

9. ¿Puedes proporcionar un ejemplo de un proyecto en el que hayas utilizado WebSockets y cómo lo implementaste?

Si una vez utilicé WebSockets con PHP 5 para crear un chat en tiempo real en mi sitio web de servicios técnicos. Desde entonces, no he tenido la oportunidad de implementar esta tecnología en nuevos.



10. ¿Cuál ha sido tu experiencia previa en liderar proyectos de tecnología de la información?

En cuanto a mi experiencia en liderazgo de proyectos de tecnología de la información, he de admitir que aún no he tenido la oportunidad de liderar ninguno. Sin embargo, he llevado a cabo mis propios proyectos y he estudiado temas relacionados con la gestión de equipos. He realizado varios cursos en este ámbito, incluyendo:

1. [Liderazgo y trabajo en equipo](#)
2. [Fundamentos de gestión de equipos IT y programación](#)

3. [Cómo desarrollar la inteligencia emocional](#)
4. [Inteligencia emocional y motivación. Consigue tus objetivos](#)

Además, he escrito varios **artículos sobre temas de gestión**, incluyendo:

1. [Origen e historia del Management](#)
2. [La vida y el legado de Mary Parker Follett: influencia olvidada en la gestión moderna](#)
3. [El Síndrome de Procusto: Cómo evitar que afecte el ambiente laboral](#)

11. ¿Cómo describirías tu estilo de comunicación y cómo te relacionas con otros miembros del equipo en un entorno de trabajo?

En cuanto a mi estilo de comunicación, considero que es respetuoso y empático. Soy consciente de que en un ambiente de equipo de alto rendimiento y con plazos ajustados, pueden surgir situaciones que debemos evitar. Por ello, creo que es esencial mantenerse enfocado en la culminación de tareas, pero también estar dispuesto a ayudar a los compañeros si surge la necesidad. Sin embargo, si hay compañeros con actitudes hostiles, trato de evitarlos y aplico la inteligencia emocional para mantener un ambiente laboral saludable.

Asimismo, he optimizado procesos para lograr una colaboración mutua y beneficiosa para el proyecto. En mi empleo anterior, cuando vi que mis compañeros del back-end estaban ocupados con la base de datos, propuse una solución para agilizar el trabajo.

Creé modelos de datos para las interfaces de la App basados en supuestos endpoints del back-end, lo que permitió mostrar el progreso del proyecto al cliente antes de contar con la información real.

De esta manera, facilité a mis compañeros del back-end enfocarse en solicitudes críticas al ocuparme de la creación de los modelos de datos mientras estaban ocupados.

Mis compañeros del back-end, utilizaban mis modelos de datos como base para desarrollar los endpoints necesarios con información real, lo que mejoró la eficiencia del equipo.

12. ¿Cuánta gente has tenido a tu cargo?

Ninguno no he tenido la oportunidad de ser líder o manager aún

13. ¿Cómo manejas los plazos y la asignación de recursos en un proyecto de TI?

En mi carrera he adquirido experiencia con este tema, en mi empleo anterior en Epidata, utilizamos la metodología Scrum, que incluía la utilización de un backlog y sprints semanales, con un deploy cada viernes.

Esta metodología nos permitió tener una visión clara de los objetivos del proyecto y asegurarnos de que se avanzaba de manera eficiente y según lo planificado.

Aunque la planificación del backlog era responsabilidad de mi manager y se definía en conjunto con los requerimientos del cliente, pude participar activamente en el seguimiento y monitoreo del progreso

Además, de mi experiencia con Scrum he realizado este curso: [Scrum Master + La Revolución Ágil + Liderar Equipos Scrum](#)

NOTA:

Gracias por la oportunidad de compartir mis conocimientos y experiencia, espero haber respondido de manera clara y satisfactoria a sus preguntas.

Si necesitan más información sobre mi experiencia o cualquier otra consulta, por favor no duden en contactarme.

Mas de mi experiencia en:

<https://dennysjmarquez.dev/> , <https://www.linkedin.com/in/dennysjmarquez/>