Jose Arana

De

Java Developer

Summary

Professional who likes researching and learning new IT technologies, so I can get better and improve my skills at any time, this way I can support my friends, partners & co-workers at all to achieve our common goals and make things work out in the firm. I consider myself as someone who never gives up and likes doing his best at any project I take on; I face every project of any kind in a faithful, encouraged and in a fearless way, so now I can say these 3 qualifyings have absolutely helped me to finish all projects in a successfully way and achieve my targets so as the firm.

Working History

Java BE Engineer
Netatech – Bank discover financial services
2022 Nov – 2023 Jun

Netatech is a global consultancy tech firm that gives the chance to work in a remote fashion for lots of prjects.

Project PCF2OCP
Main responsibilities included:
Java Engineer

- Receive and work on user stories preassigned, so the goal was to migrate existing apps running on PCF to OCP, for that we had to work jointly with K8s and OCP as well as other tools, so we can migrate such apps.
- Work throughout all phases of the software development process independently as well as work jointly with other team members
- Collaborate with architects, product management, engineering, QE and Operations teams to develop innovative solutions alignment to development goals and principles
- Tools Swagger, Java11, STS, Maven, OCP, Kubernetes, Jenkins. ArgoEvent, Nexus, Aws.

Java BE Developer
TrueLogic.
2021 Mar – 2022 Jul

IT company in charge of identifying new challenges across their different kind of clients and assign an IT resource.

Project Data Lake
Main responsibilities included:

Java Developer / BI Engineer

- Participation in charge of identifying and generating the design in terms of infrastructure to support the whole Data Lake/ETL project through microservices.
- Identify the whole business process on how the data was gathered through surveys across different countries in order to catch customer criteria, to be later extracted, processed and saved into a relational database at first instance, since it had to be later aggregated and be shown in a dashboard for metrics purposes.
- Development and deploy of Java microservices to accomplish the functional data flow explained before, through AWS services with auto CI and CD support.
- Development and deploy of java components to be run as batch in an async fashion, these components were in charge of loading data as well as movin data from db instance to db instance according to ETL design and were also in control of aggregatin data.
- Develop Materialized views to save data to be later used to save time in terms of performance.
- Approve PR to merge new code into dev branch, so as merge dev into prod branch.
- Technologies: *Java 11, Git, Maven, TFS, Spring boot 2.x, Spring Data, Spring REST, STS, Postgres, Redshift, S3, CodeBuild, CodeDeploy, CodeCommit, CodeRepo, Amazon EC2.*
- Methodologies: Scrum, Agile.

Java Architect

Kruger Corp.
2018 Oct – 2020 Nov.

IT company that offers development services for all types of projects, based on a paradigm that allows them to solve problems and achieve results in the right time, with support for all types of technology, following high-level standards as well as principles based on agile methodologies.

Project Park and Go
Main responsibilities included:
MSA Architect

- Identification of the business model, associated with the parking process, in order to identify the business process, sub-processes, business use cases, specifications, basic and alternate flows for the ParkAndGo project, StartUp of the Kruger company.
- Identification of Functional Requirements exposed in natural language in order to list the functionalities that the mobile application will support in the ParkAndGo project.
- Identification and specification of use cases of Analysis of the Mobile Application for the ParkAndGo project, specifying actors, basic and alternate flows, prototypes and delegation of services for all the functionalities supported in the mobile application of the ParkAndGo project.
- Design of the identified services responsible for supporting the functionality specified in the Analysis documents, generating the artifacts associated with this design stage, such as the RESTful service contracts with Swagger 2.0 and OpenApi 3.0.
- Development and maintenance of RESTful services, under an MSA context, through Java and framework with support for the development of microservices such as Micronaut, generating necessary artifacts for the packaging of components.

- Integration with Amazon Cognito for the federation of Identity Providers used in the project implementation.
- Deployment of components implemented under a container-based architecture using Docker Swarm in OnPremise Infrastructure.
- **Support to functional and comprehensive tests, before the solution is deployed to Production, as well as the control of tests for clients of the application based on Android Mobile.**
- Support for the Promotion of components to the Production environment, considering the dependencies and configurations per environment.
- Design of the ParkAndGo Project Architecture based on MSA, identifying components, platforms, messaging and integration against other components.
- Registration and Management of the LifeCycle of the APIs implemented for the governance of the project.
- Train the development team in the construction of MSA components, as well as in the definition of standards.
- Technologies: Java, Micronaut, AWS Cognito, KeyCloak 2.0, Docker Swarm, STS, OpenApi 3.0, Swagger 2.0, Bitbucket, Jenkins, **AWS**, Couchbase, Git.
- Methodologies used: SCRUM, Kanban.

Project PR16
Main responsibilities included:
MSA Specialist

- Identification of the business model, associated with the referral process, with the aim of identifying the business process, sub-processes separated by context and generating business use cases, specifications, basic and alternate flows for the PR16 project in the COES SINAC client.
- Identification of Functional Requirements exposed in natural language in order to list the functionalities that will be supported by the APIs to implement service level.
- Identification and specification of Service Use Cases that expose functionality, on which the responsibility of supporting the functionalities separated by context is delegated, specifying Actors, basic and alternate flows, security Preconditions and Postconditions.
- Design of the services (components) identified by Service Use case, generating artifacts such as Service contracts for RESTful APIs with Swagger 2.0 and OpenApi 3.0 for the components that expose functionality and design documents, specifying the flow that each component will support, as well as the discovery of new components that do not expose functionality but that interact in the flow of each service specification, in order to generate design artifacts of these new components, in addition to supporting error handling for each component and a design based on layers according to the component type to implement.
- Implementation and maintenance of the identified services, under an SOA Architecture, in App Connect Enterprise 11, the IBM bus, which allows to expose in an integration bus different services requested by COES to be consumed by its different agents regardless of the protocol, context , technology and platform. These APIs will be exposed on the IBM bus, so then the received messages are mapped, composed and transformed against the SOAP legacies of COES, as well as support for messaging

between applications and components through Kafka.

- Integration of OAuth 2.0 as an Authorization protocol in order to protect the RESTful APIs exposed on the bus, allowing access to these resources only to registered users who present a valid token from a trusted source.
- Design and implementation of OAuth 2.0 specifications, to integrate Authentication and Authorization flows to the COES security schemes and federate the user and application repository in a single central point.
- Integration of the source code, as well as configuration files for this project to a bitbucket repository, thus allowing the project to be anchored to a single source and from this generate different branches for the construction of project components and support for a continuous integration scheme.
- Release of components (RESTful APIs) built, under MSA context through Docker Swarm 2.0, for the orchestration of services in the cluster, prior to the generation of images.
- Initialization and configuration of the cluster in the Development and Production environments, as well as the registration and assignment of the nodes that will form part of the respective cluster by environment.
- Implementation of a monitoring layer, through ELK, allowing the generation of traces with the information of the transactions generated in the APIs, for subsequent analysis and monitoring.
- **Support for functional and unit tests per environment, before promoting the components to production.**
- Design of the Architecture of the PR16 Project under MSA context, identifying components, platforms, messaging through Kafka and integration against other components.
- Promotion of components to the respective development and production environments, after setting dependencies per environment.
- Responsible for the specification of productive infrastructure, high availability scheme, through Nginx, and distribution of components for the production environment.
- Generation and installation of certificates in the balancers of the solution for the productive environment.
- Preparation and automation of tasks necessary to prepare the solution to a scheme with support for continuous integration under a DevOps scheme.
- Technologies:. Java, *Swagger 2.0, OpenApi 3.0, Bitbucket, Amazon Cognito, OAuth 2.0, OIDC, JWT, IBM Toolkit, ESQL, IBM App Connect Enterprise, Postman, Gitbash, Docker Swarm 2.0, Kafka, ELK, Nginx, Git bash, Portainer.*
- Methodologies used: SCRUM, Kanban.

Project PR15

Main responsibilities included:
MSA Architect

- Identification of the business model, associated with the generation process, in order to identify the business process, sub-processes separated by context and generate business use cases, specifications, basic and alternate flows for the PR15 project in the COES SINAC client.
- Identification of Functional Requirements, exposed in natural language in order to list the functionalities that will be supported by the APIs to implement service level.
- Identification and specification of Service Use Cases that expose functionality, on

which the responsibility of supporting the functionalities separated by context is delegated, specifying Actors, basic and alternate flows, security Preconditions and Postconditions.

- Design of the services (components) identified by Service Use case, generating artifacts such as Service contracts for RESTful APIs with Swagger 2.0 and OpenApi 3.0 for the components that expose functionality and design documents, specifying the flow that each component will support, as well as the discovery of new components that do not expose functionality but that interact in the flow of each service specification, in order to generate design artifacts of these new components, in addition to supporting error handling for each component and a design based on layers according to the component type to implement.
- Implementation and maintenance of the identified services, under an SOA Architecture, in App Connect Enterprise 11, the IBM bus, which allows to expose in an integration bus different services requested by COES to be consumed by its different agents regardless of the protocol, context , technology and platform. These APIs will be exposed on the IBM bus, so then the received messages are mapped, composed and transformed against the SOAP legacies of COES, as well as support for messaging between applications and components through Kafka.
- Integration of OAuth 2.0 as an Authorization protocol in order to protect the RESTful APIs exposed on the bus, allowing access to these resources only to registered users who present a valid token from a trusted source.
- Design and implementation of OAuth 2.0 specifications, to integrate Authentication and Authorization flows to the COES security schemes and federate the user and application repository in a single central point.
- Integration of the source code, as well as configuration files for this project to a bitbucket repository, thus allowing the project to be anchored to a single source and from this generate different branches for the construction of project components and support for a continuous integration scheme.
- Release of components (RESTful APIs) built, under MSA context through Docker Swarm 2.0, for the orchestration of services in the cluster, prior to the generation of images.
- Initialization and configuration of the cluster in the Development and Production environments, as well as the registration and assignment of the nodes that will form part of the respective cluster by environment.
- Implementation of a monitoring layer, through ELK, allowing the generation of traces with the information of the transactions generated in the APIs, for subsequent analysis and monitoring.
- **Support for functional and unit tests per environment, before promoting the components to production.**
- Design of the Architecture of the PR15 Project under MSA context, identifying components, platforms, messaging through Kafka and integration against other components.
- Promotion of components to the respective development and production environments, after setting dependencies per environment.
- Responsible for the specification of productive infrastructure, high availability scheme, through Nginx, and distribution of components for the production environment.
- Generation and installation of certificates in the balancers of the solution for the productive environment.
- Preparation and automation of tasks necessary to prepare the solution to a scheme

with support for continuous integration under a DevOps scheme.

- Technologies:. Java, *Swagger 2.0, OpenApi 3.0, Bitbucket, Amazon Cognito, OAuth 2.0, OIDC, JWT, IBM Toolkit, ESQL, IBM App Connect Enterprise, Postman, Gitbash, Docker Swarm 2.0, Kafka, ELK, Nginx, Git bash, Portainer,* ***Cypress.***
- Methodologies used: SCRUM, Kanban.

Project DevOps Implementation
Main responsibilities included:
Devops Specialist

- Identification of the business process, actors, areas involved, tasks, roles and responsibilities associated with the software construction process in the COES client for the DevOps Implementation project.
- Specification of the main Pain Points, areas of concern, as well as bottlenecks in the software construction process in a Brown Paper workshop, through which techniques were implemented to ground the context of these problems.
- Identification of main workflows on which DevOps will be implemented, generating the workshop's own artifacts.
- Specification of each Workflow, considering areas involved, tools, roles and stages of the development process, generating according to each specification, a set of responsibilities in each area.
- Specification and assignment of ways of working, according to the capacity of the team and the distribution of tasks in the construction of components, thus achieving a more agile work style.
- Specification of the automation pipelines that will be supported by each workflow, these pipelines will be built according to their specification.
- Integration of the project on which DevOps will be implemented with the bitbucket repository, with the aim of starting the automation process.
- Construction of the pipelines that will allow, from the moment a change is propagated in the repository, to start the compilation process, generation of artifacts, unit tests and deployment in development, test and production environments with support for the .NET language which is the one used in the implementation of the legacies that COES uses.
- Identification of the services that will allow to implement DevOps on the identified projects, such as Azure Boards, Azure Pipelines, Azure Monitor under a cost and license.
- Monitoring of the tasks generated by the implementation, for their subsequent analysis.
- Technologies:. Git, TFS, Azure boards, Azure Monitor, Azure Pipelines.
- Methodologies used: Brown Paper, Agile

Project AWS Design
Main responsibilities included:
Java Architect

- Participation as Architect in charge of generating the design of the Architecture of an OnPremise solution with support for Oracle database, for the Development, Test and Production environments, based on AWS, for an external client. For this, the common components, volumetric size, components with licensed applications were identified and according to this, a design was generated with EC2 instances for the application

and database components, EBS for internal storage, VPC, public and private Subnets, high availability schemes for the production environment, with balancers and some availability zones, as well as support for Oracle license configurations.
- Stacks with specifications were generated to deploy this solution under a continuous integration scheme with AWS CloudFormation.
- Technologies: *Kubernetes, Amazon Elb, EC2, Route53, AZ, Cloud Formation, RDS , Replica Read, Jenkins, Ebs, CodeBuild, CodeDeploy, CodeCommit, CodeRepo, Lambda, Amazon Api Gateway, EKS*.
- Methodologies: Scrum, Agile.

SOA Architect

Hundred SAC

2018 Jun – 2017 Jun

IT consultant, who attends development projects for clients of any kind, makes use of some old practices.

Project SOA Implementation

SOA Architect

Main responsibilities included:

- Business identification, through understanding meetings to understand the process and identify use business case and workflows of Equifax Peru client, through a Domain Driven Design approach that allows us to translate and map the business sub processes in domains that are part of the business core and from these generate business artifacts.
- Evaluate the impact of the components in the project solution, as well as the type of architecture, viability, low coupling and high cohesion in such a way that a redesign in the architecture of the solution does not impact on the performance of the same.
- Design and specify Logical Architecture Documents and Data Architecture Documents in the project solution where the architecture, workflows, distribution and metrics that will support each component under an SOA architecture are defined, as well as specifying if the component will be new and / or reused.
- Verify if the contracts of the designed components support what the business functionally requires.
- Elaborate and specify RESTFUL and SOAP Web Services Contracts , specifying Request and Response parameters type for each method of each component in the project solution.
- Development and maintenance of Web Service type components under J2EE technology, design patterns and JSR 109/224/311 specifications (Web Services / JAX - WS / JAX - RS) under a RESTFUL microtechnology architecture with Spring MVC, Spring Data, Spring Batch, Spring Boot, Spring Security and Spring Cloud technologies.
- Integration of RESTFUL microservice-type components through Spring Cloud and in this way to define a distributed architecture based on RESTFUL microservices and implement High Availability, Load Balancing and Integration technologies through Feign, ConfigServer, Eureka, Ribbon and Zuul.
- Deployment and monitoring of the components through Oracle SOA Suite 12c, Weblogic Server - Oracle Sever Bus - Oracle SOA Infra for the deployment of BPEL type components through continuous integration.

Technologies: Weblogic Server - Oracle Sever Bus - Oracle SOA Infra, ConfigServer, Eureka, Ribbon, Hystrix and Zuul.

- Methodologies: Scrum

Senior Java Developer
Everis
2016 Jan – 2018 Jun

IT consultant, who attends development projects for clients of any kind, develops integration components in an EAI layer.

Project EAI Implementation
Senior Java Developer
Main responsibilities included:

- Business Identification through meetings to understand the process and identify América Móvil Client- Claro business use cases.
- Analysis of the Service Contract designed by the Architecture area in order to identify the structure of the web service to be developed as well as the inputs and outputs of the web service methods to be built.
- Analysis and Design of XSD schemes to generate the WSDL of the web services to be built.
- Development and maintenance of Java Web Services under J2EE tech, MVC patterns & JSR 109/224(Web Services/JAX WS).
- Design and implementation of granular services bus without flow for the exposure of built web services and Stored Procedures Type Components under Oracle SOA suite.
- Development and maintenance of Web Components under J2EE tech, MVC patterns & JSR 286(Portlets)/JSR 315(Servlets) using Spring MVC.
- Development and maintenance of components under J2EE tech, MVC patterns & JSR 352(Spring Batch)for processing massive objects in an asynchronous parallel way and define parameters that allow supporting rules and business oriented tasks.
- Development and maintenance of EJB 3.0 type components under J2EE technology & MVC patterns, MDB(for processing objects in parallel, massive and asynchronous way through queues and JMS transport), Session Bean(for remote access to external platforms by RMI).
- Development and maintenance of backend type middleware components with support for SOA technology(Stored Procedures, Web Services, Oracle Service Bus - BPEL, LDAP).
- Support, configuration and maintenance of continuous integration platform and DevOps practices through an integration with Jenkins Platform.
- Development and maintenance of BPEL 2.0 type component for the orchestration of OSB, WS and EIS type components(external platform adapters through JCA) with business rules support.
- Deployment of BPEL-type components in Weblogic Server - SOA infra console through JAR packages and XML deployment plans(configurable by environment & generated from project's .composite).
- Technologies: Spring Tools Suite, JDeveloper 11g, Eclipse Helios IDE, JDeveloper Fussion MiddleWare (for OSB Configuration), Spring Framework, Spring JDBC,

MyBatys Framework, Weblogic Server Application Server Oracle, BPEL Suite SOA Oracle, IBM WebSphere Application Server, Specification Java (JSR 109 / JSR 224 / JSR 286), JAX - WS RI, Soap UI, Java J2EE, Oracle DataBase 12c, SFTP, FileZilla, WinScp, Tortoisse SVN, Bizagi Process Modeler, VMWare, Log4J, AspectJ, Web Portal Server IBM, PMD (for Code Quality), Jenkins, Maven (for dependency management in Java)

- Methodologies: Scrum.

Java Developer
CTI Solution
2014 May – 2015 Dec

IT consultant, who attends development projects for clients of any kind, develops integration components in an EAI layer.

Project EAI Implementation
Java Developer
Main responsibilities included:

- Business Identification through meetings to understand the process and identify business use cases.
- Analysis of the Service Contract designed by the Architecture area in order to identify the structure of the web service to be developed as well as the inputs and outputs of the web service methods to be built.
- Analysis and Design of XSD schemes to generate the WSDL of the web services to be built.
- Development and maintenance of Java Web Services under J2EE tech, MVC patterns & JSR 109/224(Web Services/JAX WS).
- Design and implementation of granular services bus without flow for the exposure of built web services and Stored Procedures Type Components under Oracle SOA suite.
- Technologies: Spring Tools Suite, JDeveloper 11g, Eclipse Helios IDE, JDeveloper Fussion MiddleWare (for OSB Configuration), Spring Framework, Spring JDBC, MyBatys Framework, JAX - WS RI, Soap UI, Java J2EE, Oracle DataBase 12c, SFTP, FileZilla, SVN, Bizagi Process Modeler, VMWare, Log4J, AspectJ, Web Portal Server IBM, PMD (for Code Quality), Jenkins, Maven (for dependency management in Java)
- Methodologies: Scrum.

Educational History / Training Courses

AWS Training Center
AWS Developer
Currently

Cesar Vallejo
Computer Science Engineer
2009- 2014

Universidad Nacional Ingenieria
UML Developer Workshop
2014- 2014

Universidad Nacional Ingenieria

PHP Developer
2013 – 2013

CEPS
Networking Managment
2012