



“INVESTIGACION 1”

UNIVERSIDAD TECNOLÓGICA DE SAN LUIS RIO COLORADO

JAVIER VELAZQUEZ

DESARROLLO WEB INTEGRAL

FIDEL OROZCO CASTRO

IDGS9-1

24 DE SEPTIEMBRE DEL 2024

METODOLOGIAS AGILES

Según el portal (Zendesk, 2023) las metodologías ágiles pueden traducirse como un conjunto de técnicas, segmentadas en ciclos de trabajos cortos que permiten la entrega de un proyecto con mayor rapidez. Estas en comparación de las tradicionales llevan algunas diferencias como un mayor control de presupuesto en sincronización con la entrega de resultados pequeño, una mayor flexibilidad al trabajarse con ciclos de trabajo “sprint” que al mismo tiempo mejora en la gestión de riesgos y la colaboración con el cliente ya que en este tipo de metodologías la participación del cliente es esencial debido a que es tomado en cuenta de forma activa como parte del ciclo.

MODELOS DE ARQUITECTURA

1. Modelo-Vista-Controlador (MVC):

- MVC es una arquitectura ampliamente utilizada para construir interfaces de usuario.
- El modelo representa los datos y la lógica, la vista es responsable de la presentación y el controlador se encarga de las interacciones con el usuario.
- MVC proporciona separación de preocupaciones y permite un fácil mantenimiento y reutilización del código.
- Entre los marcos de trabajo más populares que implementan MVC se encuentran Ruby on Rails, Django y ASP.NET MVC.

2. Arquitectura de microservicios:

- Los microservicios consisten en dividir una aplicación en pequeños servicios independientes.
- Cada microservicio se centra en una capacidad de negocio específica y se comunica a través de protocolos ligeros como REST o colas de mensajería.

- Los microservicios favorecen la escalabilidad, la tolerancia a fallos y el desarrollo rápido mediante equipos descentralizados.
- Frameworks como Spring Boot, Node.js con Express o .NET Core permiten crear microservicios.

3. Arquitectura orientada a servicios (AOS):

- SOA es un estilo arquitectónico que enfatiza los servicios libremente acoplados.
- Los servicios son componentes autónomos y reutilizables que se comunican a través de interfaces bien definidas.
- AOS facilita la interoperabilidad, la flexibilidad y la reutilización en diferentes aplicaciones.
- Frameworks como Apache CXF y Oracle AOS Suite ayudan a implementar soluciones basadas en AOS.

4. Arquitectura en capas:

- La arquitectura en capas organiza el sistema en capas horizontales, cada una responsable de una funcionalidad específica.
- Las capas incluyen la presentación, la lógica de negocio y el acceso a los datos, garantizando la separación de preocupaciones.
- La arquitectura en capas simplifica el mantenimiento y permite actualizaciones independientes de las distintas capas.
- Frameworks como ASP.NET MVC, Django y Laravel proporcionan una base para
- construir aplicaciones en capas.

5. Arquitectura nativa en la nube:

- La arquitectura nativa en la nube está diseñada para aplicaciones desplegadas en la nube
- Aprovecha la escalabilidad, resiliencia y elasticidad que ofrecen las plataformas en la nube.
- A menudo se utiliza la contenedorización con Docker y la orquestación con Kubernetes.
- Marcos como Spring Cloud, AWS Lambda, Azure Functions y Google Cloud Functions apoyan el desarrollo nativo en la nube.

6. Arquitectura sin servidor:

- La arquitectura sin servidor permite construir aplicaciones sin gestionar la infraestructura subyacente.
- Aprovecha las funciones como servicio (FaaS) y los modelos informáticos basados en eventos.
- Los desarrolladores se centran en escribir funciones que responden a eventos, reduciendo la sobrecarga operativa.
- Marcos como AWS Lambda, Azure Functions y Google Cloud Functions simplifican el desarrollo sin servidor.

Fuente: (Pandey, 2023)

HERRAMIENTAS PARA EL MODELADO DE SOFTWARE

Según la información tomada del artículo web (Lago, 2022) algunas de las herramientas mas utilizadas para el uso de modelado de software suelen ser: Rational Rose, Visual Paradigm, Visio de Microsoft, StarUML Enterprise Architect y Magic Draw esto sin mencionar la gran facilidad de manejar y crear por propias manos el lenguaje de modelado de software.

FRAMEWORKS MAS USADOS (BACKEND Y FRONTEND)

BackEnd	FrontEnd
ASP.NET Core	React
Django	Angular
Laravel	VueJS
Express.js	JQuery
CakePHP	EmberJS

Fuentes: (Gadhavi, 2024), (Dhaduk, 2024)

CONCLUSIONES

Como conclusión queda decir que el mundo de la programación al menos en mi punto de vista no es solamente sentarte a realizar código mediante la lógica sino que al mismo tiempo todo lleva un orden para que los proyectos a realizar no terminen siendo un caos, las metodologías ágiles me parecen interesantes ya que es emocionante para mi ir realizando pequeños avances que mostrar y que vayan siendo aceptados.

REFERENCIAS

Dhaduk, H. (1 de Febrero de 2024). *Simform*. Obtenido de <https://www.simform.com/blog/best-frontend-frameworks/>

Gadhavi, M. (16 de Junio de 2024). Obtenido de <https://radixweb.com/blog/best-backend-frameworks>

Lago, N. (31 de Mayo de 2022). *SaasRadar*. Obtenido de <https://saasradar.net/modelado-de-software/>

Pandey, P. (11 de Julio de 2023). *Medium*. Obtenido de <https://medium.com/@publicapplicationcenter/tutorial-notes-common-software-architecture-frameworks-1a9915e1d806>

Urwin, M. (19 de Diciembre de 2023). *builtin*. Obtenido de <https://builtin.com/articles/smart-device>

Zendesk. (2023 de Febrero de 2023). Obtenido de <https://www.zendesk.com.mx/blog/metodologia-agil-que-es/>