**Onboarding Practico**

|  |  |
| --- | --- |
| **Información del Recurso** | |
| **Perfil Back o Front** | Backend |
| **Nombre:** | Victor Maldonado |
| **Email personal:** | [victor\_fab92@hotmail.com](mailto:victor_fab92@hotmail.com) |
| **Email banco:** | vimaldon@pichincha.com |
| **Computador (código)** | 398263 |
| **Tribu o Célula** | **MEDIOS DE PAGO/PROYECTO TARJETAS DE CREDITO** |
| **Información del Documento** | |
| **Tutor:** | CHRISTIAN ALEXANDER YEPEZ CARRERA |
| **Fecha Inicio OnBoarding** | 02-08-2022 |
| **Fecha Fin Onboarding** | 09-08-2022 |
| **Link Repositorio** | https://github.com/vmh82/BPichincha-Onboarding.git |

# Informe de parte de Colaborador

Contenido

[Introducción Onboarding 4](#_Toc110924498)

[Onboarding 02-08-2022 4](#_Toc110924499)

[Preparar ambiente para desarrollo de caso práctico 4](#_Toc110924500)

[Onboarding 03-08-2022 4](#_Toc110924501)

[Revisión Arquetipo Entregado 4](#_Toc110924502)

[Historias de usuario 4](#_Toc110924503)

[Identificación de dominio 5](#_Toc110924504)

[Revisión HU-Carga Inicial de clientes 5](#_Toc110924505)

[Revisión HU-Carga Inicial de Marcas 5](#_Toc110924506)

[Revisión HU-Carga Inicial de Ejecutivos 5](#_Toc110924507)

[Onboarding 04-08-2022 6](#_Toc110924508)

[Desarrollo de componentes y utilitarios 6](#_Toc110924509)

[HU-Administración de clientes 6](#_Toc110924510)

[HU-Administración de Marcas 6](#_Toc110924511)

[HU-Administración de Vehiculos 7](#_Toc110924512)

[HU-Administración de Patios 7](#_Toc110924513)

[Onboarding 05-08-2022 8](#_Toc110924514)

[HU-Asignación de clientes 8](#_Toc110924515)

[HU-Solicitud de Crédito 8](#_Toc110924516)

[Pruebas de Integración 9](#_Toc110924517)

[Onboarding 08-08-2022 9](#_Toc110924518)

[Observaciones Generales 9](#_Toc110924519)

[Compromiso 9](#_Toc110924520)

[Anexos 10](#_Toc110924521)

[Test simulación Carga general de documentos 10](#_Toc110924522)

[Endpoints 11](#_Toc110924523)

[Administración de Clientes 11](#_Toc110924524)

[Pruebas Unitarias Clientes 11](#_Toc110924525)

[Pruebas de Integración Clientes Service 12](#_Toc110924526)

[Administración de Marcas 13](#_Toc110924527)

[Pruebas Unitarias Marcas 13](#_Toc110924528)

[Administración de Vehículos 13](#_Toc110924529)

[Administración de Patios 14](#_Toc110924530)

[Asignación de Cliente 14](#_Toc110924531)

[Solicitud de crédito 15](#_Toc110924532)

[Pruebas Unitarias solicitud de crédito 15](#_Toc110924533)

[Pruebas de Integración solicitud de crédito 16](#_Toc110924534)

# Introducción Onboarding

Se procede con el desarrollo del caso práctico correspondiente a “Solicitud de crédito auto” considerando las siguientes historias de usuario

|  |
| --- |
| HU-Carga Inicial de Clientes |
| HU-Carga Inicial de Marcas |
| HU-Carga Inicial de Ejecutivos |
| HU-Administrar Clientes |
| HU-Administrar patios de autos |
| HU-Administrar vehículos |
| HU-Asignación de clientes |
| HU-Generar solicitud de crédito |

# Onboarding 02-08-2022

Preparar ambiente para desarrollo de caso práctico

Durante esta etapa se procede a configurar el entorno de desarrollo de manera local, con las siguientes herramientas:

* Microsoft Visual Studio Community 2022
* SQL Server 2019
* Postman
* Lenguaje base C#

Posterior a esto se revisa la documentación oficial de Microsoft sobre TDD y como llevarla a la practico en .NET

# Onboarding 03-08-2022

## Revisión Arquetipo Entregado

Durante esta etapa se realiza una revisión sobre el arquetipo entregado con el objetivo de identificar la arquitectura utilizada, distribución de capas, test unitarios, test de integración, y lineamientos del proyecto

Seguido a esto se procede con la generación de preguntas con el objetivo de despejar las dudas sobre la arquitectura y distribución de capas antes de iniciar con el desarrollo.

## Historias de usuario

Posterior a esto se procede con la revisión de historias de usuario con el objetivo de identificar el dominio del negocio, consideraciones previas al desarrollo y alcance.

## Identificación de dominio

Dentro de este apartado se procede a identificar el problema a solventar, actores involucrados en el dominio de negocio y reglas de validación que se deben considerar durante el desarrollo

## Revisión HU-Carga Inicial de clientes

Se procede con la revisión de la historia de usuario correspondiente a la carga inicial de clientes, para lo cual se inicia con la creación del archivo Clientes.csv

A continuación, se procede a crear una prueba unitaria que simule la carga inicial de clientes, utilizando como marco de trabajo TDD y Nunit, el objetivo de esta prueba es identificar los objetos y estructuras necesarios que se requerirán para iniciar este desarrollo y simular el escenario sobre el cual trabajará esta funcionalidad, dentro de esta etapa se considera la creación de estructuras en base de datos, creación de contextos y mapeo de objetos de base de datos a entidades.

## Revisión HU-Carga Inicial de Marcas

Se procede con la revisión de la historia de usuario correspondiente a la carga inicial de marcas, para lo cual se inicia con la creación del archivo Marcas.csv

A continuación, se procede a crear una prueba unitaria que simule la carga inicial de marcas, utilizando como marco de trabajo TDD y Nunit, el objetivo de esta prueba es identificar los objetos y estructuras necesarios que se requerirán para iniciar este desarrollo y simular el escenario sobre el cual trabajará esta funcionalidad

## Revisión HU-Carga Inicial de Ejecutivos

Se procede con la revisión de la historia de usuario correspondiente a la carga inicial de ejecutivos, para lo cual se inicia con la creación del archivo Ejecutivos.csv

A continuación, se procede a crear una prueba unitaria que simule la carga inicial de ejecutivos, utilizando como marco de trabajo TDD y Nunit, el objetivo de esta prueba es identificar los objetos y estructuras necesarios que se requerirán para iniciar este desarrollo y simular el escenario sobre el cual trabajará esta funcionalidad.

Revisar [Anexo Test de Carga Inicial de Documentos](#_Test_simulación_Carga), en el presenta la carga inicial de documentos al iniciar la aplicación, considerando las reglas de negocio durante el proceso de carga.

# Onboarding 04-08-2022

## Desarrollo de componentes y utilitarios

Dentro de esta etapa se procede con el desarrollo de componentes y utilitarios para manejar las respuestas, mapear entidades, constantes, contextos y relaciones que manejará la aplicación.

## HU-Administración de clientes

* Se aplica TDD para identificar, clases y estructuras necesarias para llevar a cabo el desarrollo, de igual manera se realizan las pruebas unitarias para simular proceso de Crud de la administración de clientes utilizando InMemory Database de EntityFramework Core.
* Creación de interfaces IRepository y IService e implementación para simular el CRUD de administración de clientes, los procesos son:
  + Consultar
  + Crear
  + Eliminar
  + Actualizar
* Creación de Dtos para envío y devolución de información
* Creación colección postman para pruebas

## HU-Administración de Marcas

* Se aplica TDD para identificar, clases y estructuras necesarias para llevar a cabo el desarrollo, de igual manera se realizan las pruebas unitarias para simular proceso de Crud de la administración de clientes utilizando InMemory Database de EntityFramework Core.
* Creación de interfaces IRepository y IService e implementación para simular el CRUD de administración de clientes, los procesos son:
  + Consultar
  + Crear
  + Eliminar
  + Actualizar
* Creación de Dtos para envío y devolución de información
* Creación colección postman para pruebas

## HU-Administración de Vehiculos

* Se aplica TDD para identificar, clases y estructuras necesarias para llevar a cabo el desarrollo, de igual manera se realizan las pruebas unitarias para simular proceso de Crud de la administración de clientes utilizando InMemory Database de EntityFramework Core.
* Creación de interfaces IRepository y IService e implementación para simular el CRUD de administración de clientes, los procesos son:
  + Consultar
  + Crear
  + Eliminar
  + Actualizar
* Creación de Dtos para envío y devolución de información
* Creación colección postman para pruebas

Ver Anexos:

[Endpoint Administración Cliente](#_Administración_de_Clientes)

[Pruebas Unitarias Cliente](#_Pruebas_Unitarias_Clientes)

[Pruebas Integración ClienteService](#_Pruebas_de_Integración)

## HU-Administración de Patios

* Se procede con la creación de clases y estructuras necesarias para llevar a cabo el desarrollo, de igual manera se realizan las pruebas unitarias para simular proceso de Crud de la administración de clientes utilizando InMemory Database de EntityFramework Core.
* Creación de interfaces IRepository y IService e implementación para simular el CRUD de administración de clientes, los procesos son:
  + Consultar
  + Crear
  + Eliminar
  + Actualizar
* Creación de Dtos para envío y devolución de información
* Creación colección postman para pruebas

Ver Anexos:

[Endpoint Administración Patios](#_Administración_de_Patios)

# Onboarding 05-08-2022

## HU-Asignación de clientes

* Se procede con la creación de clases y estructuras necesarias para llevar a cabo el desarrollo, de igual manera se realizan las pruebas unitarias para simular proceso de Crud de la administración de clientes utilizando InMemory Database de EntityFramework Core.
* Creación de interfaces IRepository y IService e implementación para simular el CRUD de asignación de clientes, los procesos son:
  + Consultar
  + Crear
  + Eliminar
  + Actualizar
* Creación de Dtos para envío y devolución de información
* Creación colección postman para pruebas

Ver Anexos:

[Endpoint Asignación de clientes](#_Asignación_de_Cliente)

## HU-Solicitud de Crédito

* Se aplica TDD para identificar, clases y estructuras necesarias para llevar a cabo el desarrollo, de igual manera se realizan las pruebas unitarias para simular proceso de Crud de la administración de clientes utilizando InMemory Database de EntityFramework Core.
* Creación de interfaces IRepository y IService e implementación para simular el CRUD de administración de clientes, los procesos son:
  + Consultar
  + Crear
* Creación de Dtos para envío y devolución de información
* Creación colección postman para pruebas

Ver Anexos:

[Endpoint solicitud de crédito](#_Solicitud_de_crédito)

[Pruebas unitarias solicitud de crédito](#_Pruebas_Unitarias_solicitud)

[Pruebas de integración solicitud de crédito](#_Pruebas_de_Integración_1)

## Pruebas de Integración

Se inicia con el proceso de pruebas de integración debido a que estas pruebas son más costosas y requieren más tiempo de prueba, dentro de este apartado se buscar verificar que los componentes trabajen de manera conjunta y respondan a las reglas, validaciones y resultados esperados, considerando que durante el desarrollo no se hay roto alguna pieza de código durante del desarrollo, para estas pruebas se hace uso de **WebApplicationFactory**

# Onboarding 08-08-2022

Dentro de esta etapa se realiza la validación de la funcionalidad solicitada, posterior se procede con el desarrollo del informe general acerca del proyecto y por consiguiente se realiza la carga del código fuente a un repositorio público GIT.

# Observaciones Generales

* Se debe detallar estado actual de colaborador versus el estado inicial luego del Onboarding.
* Se debe detallar los GAPS, o los puntos de mejora.

# Compromiso

* Definir fechas para monitorear los puntos de mejora.

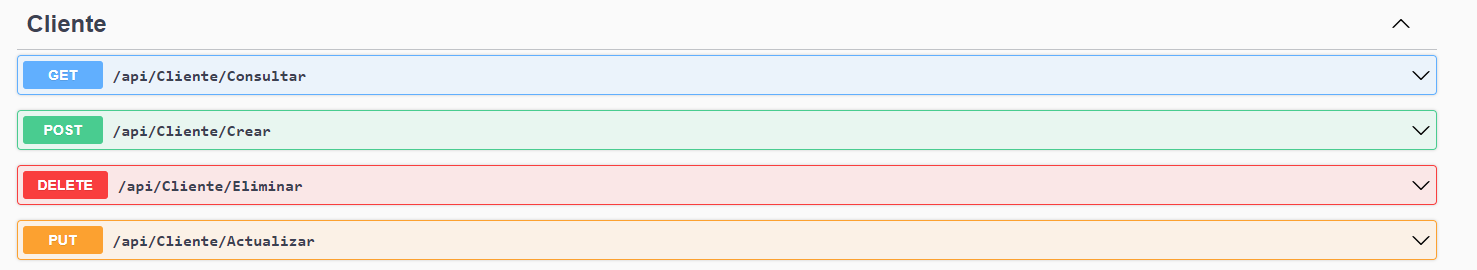
# Anexos

# [Test simulación Carga general de documentos](#_Test_simulación_Carga)



# Endpoints

## Administración de Clientes



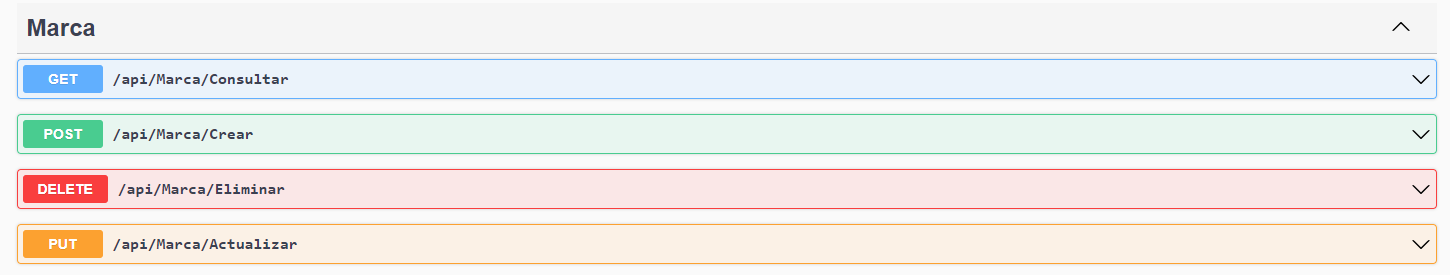
### [Pruebas Unitarias Clientes](#_Pruebas_Unitarias_Clientes)



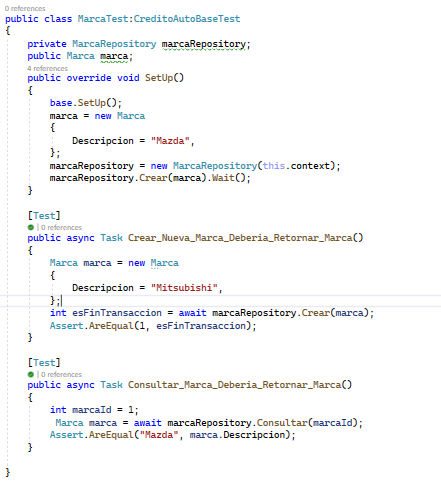
### [Pruebas de Integración Clientes Service](#_Pruebas_de_Integración)



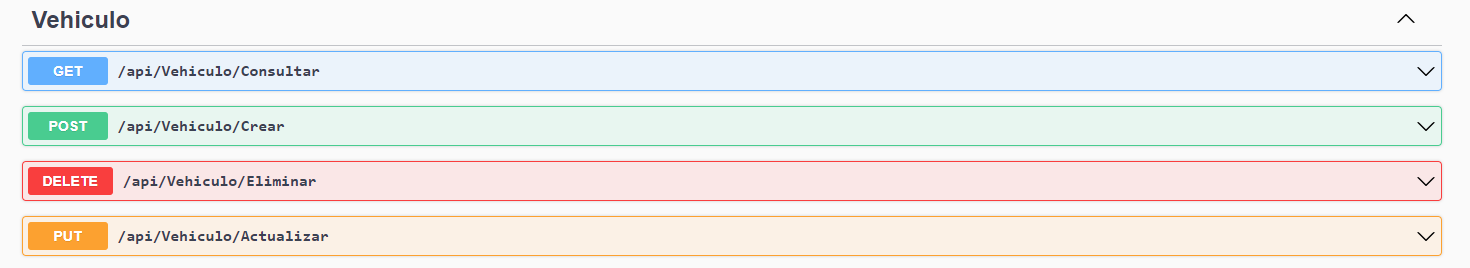
## [Administración de Marcas](#_Administración_de_Marcas)



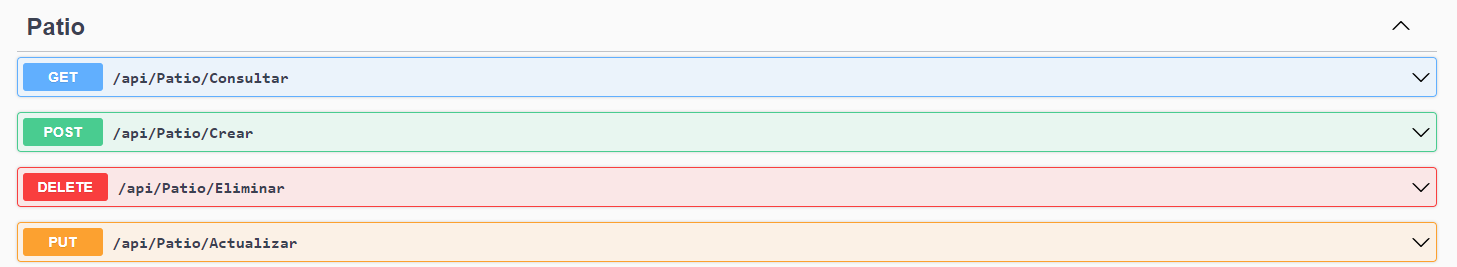
### [Pruebas Unitarias Marcas](#_Pruebas_Unitarias_Marcas)



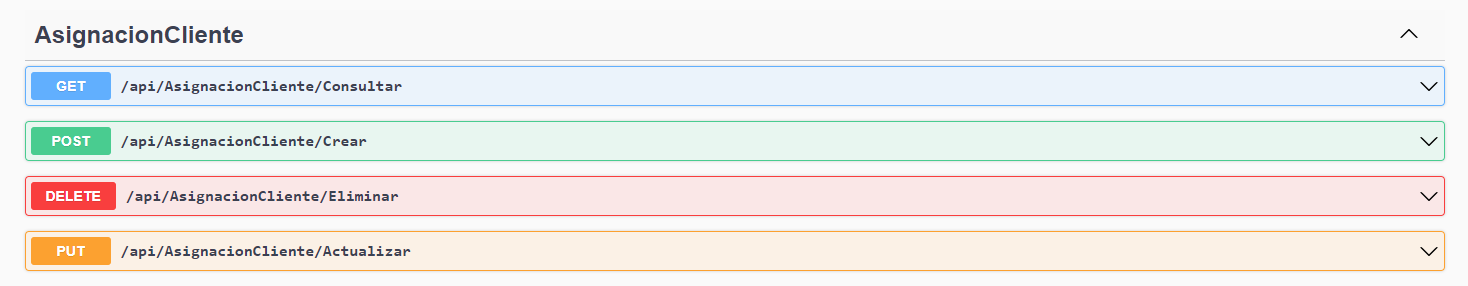
## [Administración de Vehículos](#_Administración_de_Vehículos)



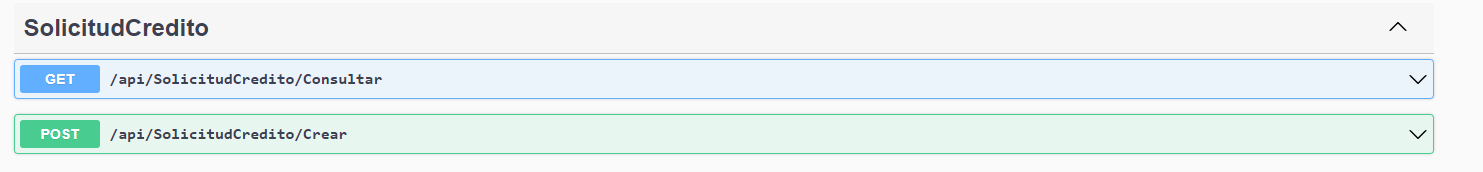
## [Administración de Patio](#_Administración_de_Patio)s



## [Asignación de Cliente](#_Asignación_de_Cliente)



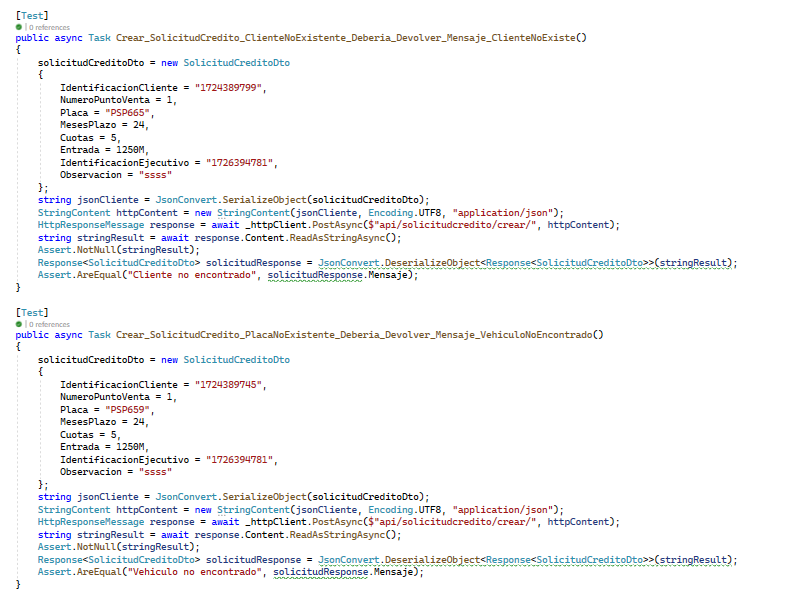
## Solicitud de crédito

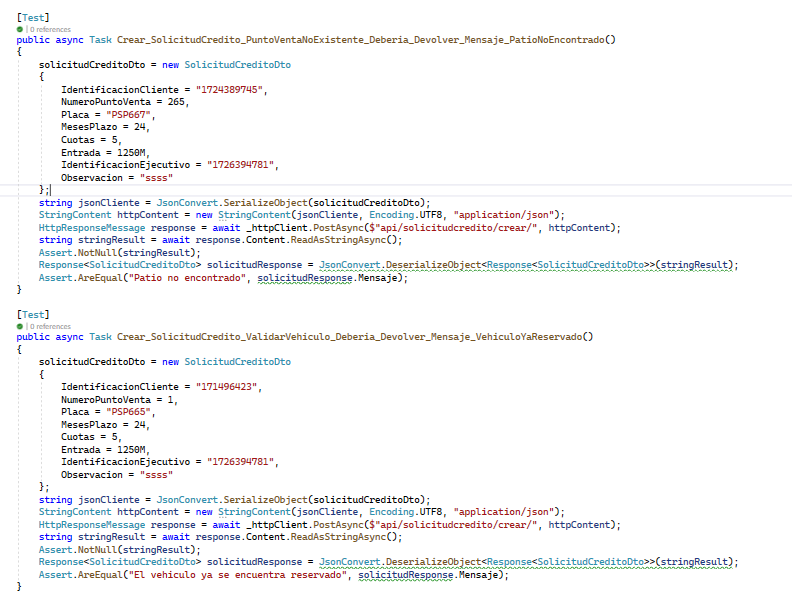


### [Pruebas Unitarias solicitud de crédito](#_Pruebas_Unitarias_solicitud)



### [Pruebas de Integración solicitud de crédito](#_Pruebas_de_Integración_1)







# Referencias.

A manera de guía se agregan ciertas referencias que se revisaron para el apartado de Test unitarios.

<https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices>

https://martinfowler.com/articles/practical-test-pyramid.html

<https://enterprisecraftsmanship.com/posts/you-naming-tests-wrong/>

https://canro91.github.io/2021/04/12/UnitTestNamingConventions/

https://canro91.github.io/2021/07/05/UnitTestingBestPractices/

https://canro91.github.io/2021/03/29/UnitTestingCommonMistakes/

<https://canro91.github.io/2020/11/02/UnitTestingTips/>

<https://canro91.github.io/assets/posts/2021-08-30-UnitTesting/UnitTesting101.pdf>

Para el apartado de arquitectura de software se agregan ciertas referencias que fueron revisadas.

<https://docs.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/ddd-oriented-microservice>

<https://blog.jacobsdata.com/2020/02/19/a-brief-intro-to-clean-architecture-clean-ddd-and-cqrs>

<https://fullstackmark.com/post/11/better-software-design-with-clean-architecture>

<https://www.ijcaonline.org/archives/volume177/number42/mohammed-2020-ijca-919923.pdf>

<https://devintxcontent.blob.core.windows.net/showcontent/Speaker%20Presentations%20Spring%202019/Clean%20Architecture%20with%20ASP.NET%20Core.pdf>

<https://docs.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/microservice-domain-model>