EMPRESA



Documento de Arquitectura de Software

OnlyCourses

Versión

1.0

Octubre, 2022

Historial de Revisiones

| **Fecha** | **Versión** | **Descripción** | **Autor (es)** |
| --- | --- | --- | --- |
| 10/19/2022 | 1.0 | Primera versión del documento de arquitectura de software, donde se detallarán las tecnologías a usar y la arquitectura elegida para el proyecto | Ashel Vasquez,  Akira Alvarado,  Brian Chiara,  Diego Zavala,  Miguel Soria,  Raquel Gonza,  Ysaac de la Cruz |
|  |  |  |  |
|  |  |  |  |

ÍNDICE

[**1. Introducción**](#_g3yu9jq6n80i) **4**

[1.1 Alcance](#_njoudphix93t) 4

[1.2 Definiciones, acrónimos y abreviaturas](#_huuxpq932upu) 4

[1.3 Resumen](#_85chbc4rvato) 4

[**2. Tecnologías utilizadas**](#_r0ad37ft3cn6) **5**

[2.1 Tecnologías utilizadas de manera general](#_hli8wgvyokgt) 5

[2.2 Tecnologías usadas en la parte del Frontend](#_i1vy2wz4l3og) 6

[2.3 Tecnologías usadas en la parte del Backend](#_a9pviwepny77) 7

[**3. Representación de la Arquitectura**](#_jk2qx0xk27sp) **7**

# 1. Introducción

La solución arquitectónica de la empresa OnlyCourses para su plataforma de educación online está enmarcada por la necesidad de un mecanismo de apoyo para aquellas universidades que deseen que sus estudiantes puedan tener acceso a cursos e información relacionada con el mundo laboral.

## 1.1 Alcance

Este documento proporciona un resumen general sobre la arquitectura del producto, utilizando las vistas necesarias de arquitectura para describir los diferentes aspectos del sistema. Con esto se pretende documentar las decisiones de arquitectura más significativas que han sido tomadas en cuenta en el proyecto.

## 1.2 Definiciones, acrónimos y abreviaturas

**Modelo de vistas 4+1:** Es un modelo que genera una perspectiva o visión de un conjunto de elementos del proyecto y sus relaciones. En conjunto la relación de vistas de arquitectura representa los aspectos de diseño y da como resultado la arquitectura de solución.

**API:** Del ingles Application Programing Interface. Es un conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre 2 aplicaciones siguiendo un conjunto de reglas.

**API Rest:** Una API REST es una interfaz de comunicación basada en el protocolo HTTP que permite a los sistemas de información: consultar, crear, editar y eliminar recursos a través de URLs.

**REST:** La transferencia de estado representacional (en inglés representational state transfer) o REST es un estilo de arquitectura de software para sistemas hipermedia distribuidos como la World Wide Web. El término se originó en el año 2000, en una tesis doctoral sobre la web escrita por Roy Fielding, uno de los principales autores de la especificación del protocolo HTTP y ha pasado a ser ampliamente utilizado por la comunidad de desarrollo.

## 1.3 Resumen

El Documento de Arquitectura abarca la definición de la arquitectura del producto a través de la especificación de la lógica (análisis y diseño), despliegue e implementación, también define los procedimientos del usuario a los que deberá dar soporte y el manejo que se realizará a los datos.

# 2. Tecnologías utilizadas

A continuación se detallarán las herramientas y tecnologías elegidas para el desarrollo del proyecto de OnlyCourses

## 2.1 Tecnologías utilizadas de manera general

La siguiente tabla muestra la lista de tecnologías/herramientas que se van a utilizar de manera general o global durante todo el desarrollo del proyecto.

| **Nombre** | **Descripción** | **Icono** |
| --- | --- | --- |
| Visual Studio Code | Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js y cuenta con un amplio ecosistema de extensiones para otros lenguajes y tiempos de ejecución. |  |
| GitHub | Plataforma creada para que desarrolladores suban el código de sus aplicaciones y herramientas, y que como usuario no sólo puedas descargarte la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo. |  |
| Docker | Docker es una plataforma de software que permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución. |  |
| Docker Compose[[1]](#footnote-0) | Herramienta dedicada a la orquestación local de dockers, es decir, se utiliza con el objetivo de definir y ejecutar aplicaciones Docker de varios contenedores de forma fácil y rápida. Esta definición y orquestación se lleva a cabo de forma local al interior de los containers, quienes, además, se encontrarán unidos a través de una red de Docker. |  |
| Git | Sistema de control de versiones distribuido, lo que significa que un clon local del proyecto es un repositorio de control de versiones completo. Estos repositorios locales plenamente funcionales permiten trabajar sin conexión o de forma remota con facilidad. Los desarrolladores confirman su trabajo localmente y, a continuación, sincronizan su copia del repositorio con la copia en el servidor. |  |

## 

## 2.2 Tecnologías usadas en la parte del Frontend

La siguiente tabla muestra la lista de tecnologías/herramientas que se van a utilizar durante el desarrollo de la parte del frontend de la aplicación.

| **Nombre** | **Descripción** | **Icono** |
| --- | --- | --- |
| NextJS | Framework de React para desarrollar aplicaciones Javascript de una sola página (SPA) de manera sencilla y que son fáciles de desplegar. |  |
| Redux | Librería de JavaScript que permite manejar el estado de una aplicación. También podríamos decir que es un patrón de arquitectura de datos que permite manejar estados |  |

## 

## 2.3 Tecnologías usadas en la parte del Backend

La siguiente tabla muestra la lista de tecnologías/herramientas que se van a utilizar durante el desarrollo de la parte del backend de la aplicación

| **Nombre** | **Descripción** | **Icono** |
| --- | --- | --- |
| NodeJS | Entorno de ejecución de JavaScript orientado a eventos asíncronos, diseñado para crear aplicaciones escalables fuera del navegador. |  |
| Express | Framework minimalista para Node.js que permite estructurar una aplicación de una manera ágil, proporcionandonos funcionalidades como el enrutamiento, opciones para gestionar sesiones y cookies, etc. |  |
| Sequelize | Sequelize es un ORM para Nodejs que nos permite manipular varias bases de datos SQL de una manera bastante sencilla, entre estas bases de datos podemos encontrar: mysql, sqlite, postgres, etc. |  |

# 3. Representación de la Arquitectura

La arquitectura diseñada para el proyecto de OnlyCourses, está basada en una arquitectura monolítica de 3 capas: capa de presentación, capa de lógica del negocio y capa de datos.

## 3.1 Capa de Presentación

En esta capa se presenta la interfaz de usuario con la que el usuario va a interactuar. En este caso, la interfaz se mostrará a través de un navegador web, gracias al uso de Next.js

## 3.2 Capa de Lógica de Negocio

Es el nivel medio o el nivel de la aplicación, aquí se procesa la información obtenida del nivel superior. Esto mediante las reglas del negocio. Puede modificar, añadir o eliminar la información de la capa de menos nivel. A este nivel se desarrolla utilizando Node.js en conjunto con Express, también se incluyen las API´s como las API de Stripe que se utilizaran en el proyecto.

## 3.3 Capa de Datos

También se le denomina capa de base de datos. Aquí se almacenarán los datos mediante tablas de las diferentes entidades de la aplicación. Para este proyecto el gestor de base de datos utilizado será MySQL

1. El uso de docker compose es para poder realizar una conexión a una instancia de MySQL sin la necesidad de descargar MySQL como tal y tampoco otras herramientas externas para poder conectarnos a ella como XAMPP. [↑](#footnote-ref-0)