



APOYO A PROYECTOS DE INGENIERÍA TI REQUERIDOS POR LA ACADEMIA INGLÉS PARA TODOS - IPT

Trabajo de grado para optar al título de
Ingeniero en Mecatrónica

Trabajo de Grado

Autor:
Leiner Jose Barrios Medina

Tutor:
MSc. Yara Angeline Oviedo Durango

8 de Marzo de 2022

Formando **líderes** para la
construcción de un nuevo
país en paz

APOYO A PROYECTOS DE INGENIERÍA TI REQUERIDOS POR LA ACADEMIA INGLÉS PARA TODOS - IPT

Trabajo de grado Ingeniería Mecatrónica

Autor

Leiner Jose Barrios Medina

Tutor

MSc. Yara Angeline Oviedo Durango
Universidad de Pamplona



Trabajo de grado para optar al título de Ingeniero en Mecatrónica

Universidad de Pamplona
Facultad de Ingenierías y Arquitectura
Pamplona, Colombia
8 de Marzo de 2022

Dedico este trabajo a...

A mi papá, Asmed Barrios Coronel y a mi mamá Hendy Medina Cuevas quienes me apoyaron en mi camino para convertirme en quien soy actualmente; siempre me brindaron su amor incondicional en los buenos y malos momentos.

A mis hermanas, Lisney Barrios Medina y Sharith Pimienta Medina, por su cariño y estar para mí siempre.

A mi querida abuelita Eduvilia Ojeda por brindarme mucho amor, apoyo y por creer siempre en mí.

A mis familiares y amigos, quienes aportaron mucha ayuda en mi recorrido, me aconsejaron y me apoyaron.

Agradecimientos

Agradezco a los docentes que contribuyeron en mi desarrollo y crecimiento profesional a lo largo de todo el camino.

A mis papás y a mis hermanas, quienes siempre fueron mi mayor motivación y estuvieron para mí en cada momento, dándome razones y ánimos para seguir adelante y nunca rendirme.

A mis familiares, que con su apoyo incondicional, me ayudaron cuando más lo necesité.

Y a mis amigos por apoyarme e inspirarme con su valiosa compañía.

La ingeniería es lo más cercano a la magia que existe en el mundo

-Elon Musk.

Resumen

El presente documento describe como se desarrollará la práctica empresarial como ingeniero de desarrollo de software y auxiliar en áreas de sistemas, las cuales tuvieron lugar en la ciudad de Cúcuta – Norte de Santander específicamente en la sede principal de la Academia Inglés Para Todos – IPT, donde se tuvo como objetivo la correcta funcionalidad de las diferentes plataformas digitales de la empresa tales como página web, plataforma de estudiantes, plataforma de exámenes, app, entre otras. En primera instancia se le asignaron al aprendiz pequeñas rutas de aprendizaje introductorias para la formación en áreas de desarrollo de software, adquiriendo complementación en conocimientos de frontend y backend, bases de datos, manejos de API's, entre otras áreas. Dentro de las actividades principales se necesitó la implementación de un nuevo diseño original del sitio web de la empresa y mejoras en el servidor, para que este fuera desarrollado por el equipo de ingenieros de la academia, fue fundamental la construcción de una plataforma que le permita a los estudiantes conectarse de manera eficiente con las diferentes actividades académicas que se les asignen. El desarrollo de una aplicación móvil fue uno de los proyectos más importante de la empresa, lo que se buscaba era la creación de una App que tenga finalidades académicas y sea presentada mediante una modalidad divertida para los estudiantes de diferentes edades. Dentro de las actividades realizadas por la empresa y por la Universidad Francisco de Paula Santander se encuentran las pruebas EKT (English Knowledge Test). El objetivo primordial para el equipo de ingenieros de la empresa fue garantizar el buen funcionamiento de la plataforma por la cual se realizan estas pruebas, así como el buen manejo de las bases de datos, tanto técnico como ético.

Palabras clave: App, Backend, Base de datos, Desarrollo de software, Frontend, Servidor.

Abstract

This document describes how the business practice was developed as a software development engineer and assistant in systems areas, which took place in the city of Cúcuta - Norte de Santander, specifically at the main headquarters of the Academia Inglés Para Todos - IPT, where the correct functionality of the different digital platforms of the company was aimed at, such as the website, student platform, exam platform, app, among others. In the first instance, they were assigned to learn small introductory learning routes for training in software development areas, acquiring complementation in knowledge of frontend and backend, databases, API management, among other areas. Among the main activities, the implementation of a new original design of the company's website and improvements in the server were needed, so that it was developed by the team of engineers from the academy, it was essential to build a platform that allows students connect efficiently with the different academics assigned to them. The development of a mobile application was one of the most important projects of the company, what was sought was the creation of an application that has academic purposes and is presented in a fun way for students of different ages. Among the activities carried out by the company and by the Francisco de Paula Santander University are the EKT tests (English Knowledge Test). The primary objective for the company's team of engineers was to guarantee the proper functioning of the platform by which these tests are carried out, as well as the proper management of the databases, both technically and ethically.

Keywords: App, Backend, Database, Software Development, Frontend, Server.

Índice general

Agradecimientos	vii
Resumen	xi
Abstract	xiii

Capítulos	Página
1 Introducción	1
1.1 Justificación	3
1.2 Objetivos	4
1.2.1 Objetivo General	4
1.2.2 Objetivos específicos	4
2 Marco Teórico	5
2.1 Ingeniería de Software	5
2.1.1 Desarrollo de software	5
2.1.2 Desarrollo web	6
2.1.3 Desarrollo móvil	6
2.2 Bases de datos	7
2.2.1 Tipos de bases de datos	7
2.2.2 Servidor	8
2.2.3 Tipos de servidores	8
2.3 Frameworks	10
2.4 Programación orientada a objetos (POO)	10
2.4.1 Clases en POO	11
2.4.2 Objetos en POO	11
2.4.3 Lenguajes en POO	11
3 Metodología	13
3.1 Plataformas académicas digitales durante la pandemia de la COVID-19	13
3.2 Formación introductoria	20
3.2.1 Capacitación inicial	20
3.2.2 Información de las plataformas	21

3.3	Ruta de aprendizaje	22
3.3.1	Primera sección de aprendizaje	22
3.3.2	Segunda sección de aprendizaje	29
3.3.3	Tercera sección de aprendizaje	34
3.4	Plataforma para pruebas EKT	52
3.4.1	Funcionalidad requerida por la plataforma	52
3.4.2	Elaboración de historias de usuarios	53
3.4.3	Ejecución del proyecto	57
3.4.4	Finalización y retroalimentación	67
3.5	IPT World - App móvil	68
3.5.1	Componentes	68
3.5.2	Núcleos	70
3.5.3	Páginas	77
3.5.4	Útiles	90
3.5.5	Retroalimentación del proyecto	94
3.6	IPT Platform	95
3.6.1	Requerimientos de la plataforma	95
3.6.2	Historias de usuario y metodología	96
3.6.3	Presentación del Release 1.0	101
3.6.4	Desarrollo del proyecto	103
3.6.5	Avances entregados	115
3.6.6	Retroalimentación y continuidad del proyecto	116
4	Resultados	117
4.1	EKT Platform	117
4.1.1	Vistas de Index	117
4.1.2	Vistas del dashboard	119
4.2	IPT World	122
4.2.1	Vistas de login	122
4.2.2	Vista principal con niveles	125
4.3	IPT Platform	127
4.3.1	Página de index	127
4.3.2	Ingreso	130
4.3.3	Dashboard	132
5	Conclusiones	135
6	Bibliografía	137

Índice de figuras

3.1	Resultados de la encuesta para la pregunta 1.	14
3.2	Resultados de la encuesta para la pregunta 2.	15
3.3	Resultados de la encuesta para la pregunta 3.	15
3.4	Resultados de la encuesta para la pregunta 4.	16
3.5	Resultados de la encuesta para la pregunta 5.	16
3.6	Resultados de la encuesta para la pregunta 6.	17
3.7	Resultados de la encuesta para la pregunta 7.	17
3.8	Resultados de la encuesta para la pregunta 8.	18
3.9	Resultados de la encuesta para la pregunta 9.	18
3.10	Comandos básicos.	24
3.11	Código con mala indentación.	26
3.12	Código con buena indentación.	26
3.13	Lenguajes de programación orientados a objetos.	28
3.14	Lenguajes de programación.	28
3.15	Instalación de MySQL.	31
3.16	Código HTML de clone de Google.	35
3.17	Código CSS de clone de Google.	36
3.18	Aplicación de viajes desarrollada en Flutter.	39
3.19	Uso de la arquitectura de software Vanilla.	41
3.20	Uso de la arquitectura de software Scooped Model.	41
3.21	Arquitectura BLoC en Flutter.	42
3.22	Arquitectura BLoC + Clean Architecture en Flutter.	43
3.23	Patrón BLoC en Flutter.	44
3.24	Código ejemplo en Javascript.	45
3.25	Código ejemplo en Typescript.	46
3.26	Herramientas de Angular.	47
3.27	Interpolación en Angular.	48
3.28	Sintáxis de interpolación en Angular.	48
3.29	Sintáxis de NgIf en Angular.	49
3.30	Sintáxis de NgIf - Else en Angular.	49
3.31	Sintáxis de NgFor en Angular.	50
3.32	Sintáxis de NgSwitch en Angular.	50
3.33	Historia de usuario 0001.	54
3.34	Historia de usuario 0002.	54
3.35	Historia de usuario 0003.	55

3.36	Historia de usuario 0004.	55
3.37	Historia de usuario 0005.	56
3.38	Historia de usuario 0006.	56
3.39	Historia de usuario - creación de usuario - Departamento académico. . .	97
3.40	Historia de usuario - seguimiento de pagos - Departamento administrativo. .	98
3.41	Historia de usuario - generación de COD - Departamento comercial. . .	99
3.42	Historia de usuario - registro de Placement Test - Departamento de marketing.	100
3.43	Roles dentro de la plataforma IPT.	101
3.44	Diagrama general del Release 1.0 con ruta crítica.	102
4.1	Vista de login - Plataforma EKT.	117
4.2	Vista de registro - Plataforma EKT.	118
4.3	Vista de olvidar contraseña - Plataforma EKT.	118
4.4	Vista de convocatorias vigentes y vencidas - Plataforma EKT.	119
4.5	Vista de instrucción y video tutorial - Plataforma EKT.	120
4.6	Vista de paso a paso para presentar la prueba EKT - Plataforma EKT. . .	120
4.7	Vista de grupos de estudiantes para presentar prueba divididos por hora - Plataforma EKT.	121
4.8	Vista de listado de estudiantes con puntaje - Plataforma EKT.	121
4.9	Vista de portada para login - IPT World.	122
4.10	Vista de ingreso - IPT World.	123
4.11	Vista de registro - IPT World.	124
4.12	Vista principales con lista horizontal de niveles - IPT World.	125
4.13	Vista inicio a mundos con avatares de cada nivel - IPT World.	126
4.14	Vista de Index - IPT Platform.	127
4.15	Vista de Index - IPT Platform.	128
4.16	Vista de Index - IPT Platform.	128
4.17	Vista de Index - IPT Platform.	129
4.18	Vista de Login - IPT Platform.	130
4.19	Vista de Register - IPT Platform.	131
4.20	Vista de Forgot-Password - IPT Platform.	131
4.21	Vista de usuarios, departamento académico - IPT Platform.	132
4.22	Formulario de creación de usuarios - IPT Platform.	133
4.23	Vista de cursos, departamento académico - IPT Platform.	133
4.24	Formulario de creación de cursos - IPT Platform.	134
4.25	Formulario de edición de cursos - IPT Platform.	134

Índice de tablas

3.1	Plataformas y servicios digitales de la academia Inglés Para Todos . . .	21
3.2	Comandos principales utilizados en Git	23

Índice de Códigos

3.1	Código .ts (TypeScript) del login - plataforma EKT	57
3.2	Código .ts (TypeScript) de la sección "Olvido de contraseña" - plataforma EKT	59
3.3	Código .ts (TypeScript) de la sección "Creación de cuenta nueva" - plataforma EKT	61
3.4	Código .ts (TypeScript) del home - plataforma EKT	63
3.5	Código .ts (TypeScript) de la sección de "Grupos EKT" - plataforma EKT	65
3.6	Código .ts (TypeScript) de la sección de "Score EKT" - plataforma EKT	66
3.7	Código .dart del core user-bloc - IPT World	70
3.8	Código .dart de la interfaz "media" - IPT World	71
3.9	Código .dart del modelo de niveles - IPT World	72
3.10	Código .dart del modelo de estación - IPT World	75
3.11	Código .dart de actividad 1 - IPT World	77
3.12	Código .dart de actividad 2 - IPT World	77
3.13	Código .dart de actividad 3 - IPT World	81
3.14	Código .dart de vista de autenticación - IPT World	82
3.15	Código .dart de vista de inicio - IPT World	83
3.16	Código .dart de vista de mapas - IPT World	84
3.17	Código .dart de vista de preniveles - IPT World	86
3.18	Código .dart de vista de premapas - IPT World	88
3.19	Código .dart de vista de registro - IPT World	89
3.20	Código .dart del tema de niveles - IPT World	90
3.21	Código .dart de los proveedores de servicios - IPT World	91
3.22	Código .dart de la lógica de rutas - IPT World	92
3.23	Código .dart de validadores - IPT World	93
3.24	Código TS de vista de Index - Plataforma IPT	103
3.25	Código TS de vista de Login - Plataforma IPT	105
3.26	Código TS de Register - Plataforma IPT	106
3.27	Código TS de Forgot-password - Plataforma IPT	108
3.28	Código TS de dashboard principal - Plataforma IPT	109
3.29	Código TS de creación de curso académico - Plataforma IPT	111

1 Introducción

Las tecnologías de la información han avanzado hasta cierto punto en el que son indispensables en la mayoría de proyectos de ingeniería, tomando como enfoque su creación, la cual es basada en un propósito general que se encarga de realizar una distinción entre un proceso limitado y procesos generales. Estudiar, diseñar y desarrollar sistemas informáticos de información como aplicaciones de software es el componente principal para la realización de diferentes proyectos de ingeniería TI, los cuáles tienen como funcionalidad el buen rendimiento de las aplicaciones en diferentes dispositivos de las personas.

Diversas empresas educativas e institutos requieren un aumento total o parcial de implementaciones tecnológicas en las diferentes metodologías educativas que éstas estén empleando. No obstante, la falta de equipos capacitados para garantizar esas implementaciones, puede estancar de manera temporal los proyectos que se requieran, por esa razón es importante que se cuente con un departamento de desarrollo de software para llevar a cabo todo tipo de novedades y objetivos por parte de los funcionarios.

Para poder darle solución a los diferentes inconvenientes que se puedan presentar, los centros educativos optan por tener objetivos fundamentales tales como el desarrollo de software requerido para conectarse de manera novedosa con los usuarios y personas en general, llevando a cabo todos los procesos que hacen parte de la funcionalidad de la empresa de manera eficiente y más eficaz; esto se logra con implementación de nuevas tecnologías para el desarrollo de software.

En este trabajo se presenta el desarrollo de varios proyectos requeridos por la empresa, con el fin de brindarle a los estudiantes y funcionarios del instituto la posibilidad de mejorar los procesos educativos y generales. Se trabajó en diferentes falencias que se venían presentando y así mismo se buscó una solución basándose en proyectos de ingeniería planteados por el departamento de tecnología. El usuario siempre necesita una buena conexión con las diferentes plataformas que se estén manejando, para ello fue necesario la implementación de actualizaciones en todas las aplicaciones ya desarrolladas, y en su caso, la implementación de nuevas; estos sistemas desarrollados anteriormente necesitaron una optimización de los servicios ofrecidos por el instituto, para la realización de ello, se partió de nuevas tecnologías usadas actualmente para que todas las tareas necesarias se pudieran desarrollar y que todos los servicios funcionaran de manera adecuada.

1.1 Justificación

Cuando se llevan a cabo proyectos académicos de buen impacto, es común observar contingencias, ya sea por falta de organización y propuestas tecnológicas que conlleven a la eficiencia y buen rendimiento de las actividades. Esta falta de implementación tecnológica ocasiona retrasos en la realización de actividades académicas, ya que generalmente se necesitan de plataformas digitales que nos permitan manejar bases de datos y monitorear los diferentes proyectos que se proponen en una academia de educación, en este caso, la Academia IPT; más que todo en tiempos de pandemia donde a la mayoría de empresas y entidades académicas les surgió la necesidad de implementar herramientas virtuales para las clases, exámenes, manejar base de datos de los estudiantes, etc. También provoca dificultades para los estudiantes a la hora de acceder a sus datos en la plataforma, pruebas de inglés, seguimiento de sus horarios de clase, calificaciones y muchos apartados más. Ahora bien, debido a que al evitar estos inconvenientes se logra el desarrollo óptimo de todas las actividades y proyectos lo que da solución a las necesidades que motivaron a la realización del contrato por parte de la entidad contratante. Resultó sumamente importante realizar una planificación previa y precisa que garantice el buen desarrollo de todos los proyectos requeridos por la empresa, en conclusión, si no se contara con ingenieros desarrolladores de software, surgiera la incertidumbre de no saber cómo llevar a cabo una organización de las plataformas digitales y claramente el desarrollo de estas, tales como páginas web, aplicaciones móviles, plataforma de clases, plataforma de exámenes y base de datos.

Supervisar, monitorear y optimizar el desarrollo del proyecto, permitió que se cumplieran con los objetivos acordados y en el tiempo acordado, ya que al seguir lo especificado en la programación de estos proyectos, se disminuyeron esas dificultades técnicas y académicas que retrasaban las didácticas de enseñanza de la academia y el buen aprendizaje de los estudiantes de la misma. Así mismo se generó una mayor productividad y organización en la empresa ya que se obtuvieron buenos rendimientos en el desarrollo de los procesos propios de la academia IPT. Por otra parte, se garantizó a la entidad, la solución de estas necesidades que motivaron al proyecto, de adecuar estas plataformas digitales correctamente.

1.2 Objetivos

1.2.1 Objetivo General

- Apoyar en el desarrollo de software general y áreas de sistemas para proyectos de ingeniería ti requeridos por departamento de tecnología de la academia inglés para todos – IPT.

1.2.2 Objetivos específicos

- Realizar de manera prolongada formación introductoria al desarrollo de software para apoyar los proyectos generales de la empresa.
- Desarrollar en conjunto plataforma y página web propia de la empresa.
- Contribuir en el desarrollo de una aplicación móvil académica en Flutter con didáctica de juegos.
- Mantener en buen funcionamiento la plataforma y base de datos para pruebas EKT de la Universidad Francisco de Paula Santander.

2 Marco Teórico

2.1 Ingeniería de Software

La ingeniería de software por ser una de las áreas mas importantes y con mayor campo en la industria computacional y en la cual se abarca una gran cantidad de campos, se ha tomado como referencia primordial para la realización de este proyecto, siendo la ingeniería de software una disciplina formada por un conjunto de métodos en los cuales se aplican el uso de herramientas, estructuras y técnicas que ayudan en la realización de programas informáticos, en el cual para dicho proceso se debe tener en cuenta todas las fases del ciclo de vida que son esenciales para el desarrollo de cualquier tipo de sistema de información. Desde el correcto análisis que se realiza previamente de la situación, el planteamiento del respectivo diseño hasta los tiempos de una eficaz implementación, pasando por las diferentes pruebas recurrentes para su correcta función. Se podría decir que la ingeniería de software es literalmente un continente donde se almacena todo el contenido, que sería el software en sí. [1]

2.1.1 Desarrollo de software

El desarrollo de software que encontramos actualmente se diferencia del resto básicamente por un par de características: la programación orientada a objetos y la división de las diferentes partes lógicas en nivel determinado de presentación, de aplicación y de ingreso y acceso a los datos. Otros parámetros concurridos de calidad, tales como la modularización, la creación de nuevas versiones y la respectiva documentación del código fuente, son propiamente referenciados a nuestro estándar igual que una gestión de proyectos enfocada detalladamente. [2]

Llevar a cabo el desarrollo de un software y definirlo es poder tomar el control de diseñar a partir de solo una descripción, pero el principal paso es tener definido un objetivo al que se quiera llegar y seguir cada uno de los modelos según su estructura, en el caso de la programación orientada a objetos nos enfocamos en conocimientos y procesos que deben ser analizados para la planificación y ejecución del diseño, lograr manejar un buen proceso, gestión, verificación del desarrollo nos permite manipular cualquier tipo de elementos para ofrecer al mundo día a día mejores productos. Ya que, debido a la alta complejidad de las aplicaciones modernas, hoy en día hay que prestar mucha atención al proceso de desarrollo mismo. Especialmente cuando nos referimos a proyectos en los que varios desarrolladores están trabajando al mismo tiempo, es importante

tener la formalización del proceso desde un principio para garantizar posteriormente una efectiva colaboración entre los diferentes desarrolladores.

2.1.2 Desarrollo web

Por desarrollo web entendemos la programación necesaria para la construcción de aplicaciones o sitios web, en la cual el mayor trabajo es poder mantenerlas impecables, en correcto funcionamiento y con buena seguridad. [3].

Aunque el desarrollo web sea un ámbito específico del desarrollo de software en general, el gran ecosistema de las tecnologías, lenguajes de programación y herramientas disponibles para la creación web es exageradamente grande. Para poder clasificar o dividir las áreas del desarrollo web se han dividido un par de términos:

Frontend

La parte encargada del diseño, composición y actividad de que una idea sea convertida en realidad, es el desarrollo web en el ámbito del cliente, es decir, en el navegador web. Las tecnologías principalmente usadas son HTML, CSS y el lenguaje de programación JavaScript.

Backend

Es la parte no visible, es decir, donde están todos los datos almacenados, es el desarrollo web en el ámbito del servidor, donde las tecnologías y lenguajes son directamente relacionadas con el sistema que opera el servidor de la página web. Los lenguajes de programación y tecnologías habituales para Back-end son PHP, NodeJS, Python, .NET, e incluye también entornos de bases de datos, como MySQL, PostgreSQL, SQL Server...

2.1.3 Desarrollo móvil

Las aplicaciones móviles hoy en día son muy fáciles de manejar gracias a la facilidad de acceso a internet, así mismo el avance de distintos tipos de teléfonos inteligentes los cuales cuentan con mejores sistemas operativos que facilitan el desarrollar aplicaciones. Son un conjunto de procesos y procedimientos involucrados en la escritura de software para dispositivos informáticos pequeños e inalámbricos, como teléfonos inteligentes y otros dispositivos portátiles. Tanto el desarrollo de aplicaciones web, como el desarrollo de aplicaciones móviles tiene sus principios en el desarrollo de software convencional. Sin embargo, existe una diferencia fundamental, y es que las aplicaciones móviles tradicionales a menudo se desarrollan específicamente para sacarle provecho a las características únicas de un dispositivo móvil en particular.

Actualmente, existen dos plataformas móviles las cuales son las más destacadas, iOS que pertenece a Apple y Android que pertenece a Google. Los teléfonos y tablets

de Apple traen precargadas sus aplicaciones esenciales, incluyen un navegador web y la App Store de Apple. Así mismo los dispositivos Android también vienen con aplicaciones preinstaladas muy similares y se pueden instalar más usando Google Play Store. [4].

2.2 Bases de datos

En la actualidad dependen más de sistemas de información por lo cual una de las mejores alternativas hoy en día es la consolidación de herramientas para guardar o almacenar toda aquella información necesaria y poder procesar de forma ágil y ordenada todos los datos, a esto se le conoce como base de datos. Las bases de datos siempre han sido sumamente importantes en el desarrollo de sistemas de software para las organizaciones. Conocer y manipular bases de datos resulta útil para un sinnúmero de usos, desde procesos de negocio, como crear una empresa emergente o un sistema de información eficiente y productivo, hasta temas que son mucho más especializados y enfocados como Big Data. Las bases de datos se podrían definir como una recolección de datos o información estructurada y bien clasificada que se encuentra almacenada generalmente en un sistema electrónico. Las bases de datos, al tener una inmensa cantidad de información ordenada, facilitan el uso y los métodos para encontrar datos de forma rápida y sencilla.

El propósito principal de las bases de datos tiene un principio con la necesidad de registrar y almacenar datos, es decir, surge a partir de ello. Por muchos años la forma preferible de hacer esto consistía en un archivo de documentos en plasmados en papel, pero rápidamente esto se volvió ineficiente, ya que lo siguiente a guardar datos es poderlos consultar fácilmente.

Hacer búsquedas puede ser un proceso muy variante, ya que si se tienen muchos tipos de datos también se va a querer buscar a partir de cualquiera de ellos. Esto es lo que fundamenta el principio teórico de las bases de datos. La facilidad de poder obtener cualquier tipo de información en cualquier momento sin ningún tipo de complicaciones. [5].

2.2.1 Tipos de bases de datos

Hay bases de datos relacionales:

- MySQL
 - SQL Server
 - Oracle
 - PostgreSQL
-

Como su nombre lo indica, estos tipos de bases de datos utilizan el modelo relacional y siempre será usarlas cuando los datos son consistentes y ya tienes algo planificado.

También existen las no relacionales: MongoDB y Redis, son conocidas como NO-SQL (Not Only SQL). Estas son un poco más flexibles en lo que a consistencia de datos respecta, y se han convertido en una opción que propone solucionar ciertas limitaciones que tiene el modelo relacional. [6].

2.2.2 Servidor

Son esenciales para internet, sería muy complejo navegar por allí o muy diferente a como lo conocemos hoy en día, por eso es muy importante contar con servidores que sean capaces almacenar y emitir de manera correcta e infinita la información que encontramos en la web. Es conocido también como un aparato informático que almacena, distribuye y suministra información. La funcionalidad de los servidores se basa en el modelo “cliente-servidor”. El cliente puede ser un ordenador y también una aplicación que requiera información del servidor para funcionar. Por lo tanto, un servidor puede ofrecer la información demandada por el cliente siempre y cuando el cliente cuente con esa autorización. Los servidores pueden ser físicos o virtuales.

Los servicios que prestan los servidores generalmente son requeridos continuamente, por esa razón la mayoría de los servidores nunca se apagan. Si un servidor deja de funcionar, eso podría causar muchos problemas a los usuarios. Por lo tanto, los servidores suelen estar programados para ser tolerantes y optimizables a fallos. [7]

2.2.3 Tipos de servidores

Servidor web

Permite almacenar y organizar el contenido de las páginas web y se lo proporciona al usuario a través del navegador web del usuario. La transmisión usualmente se realiza con http (HyperText Transfer Protocol). El http es el encargado de transmitir la información de la WWW (World Wide Web), estableciendo unos criterios semánticos y sintácticos para que el ordenador solicitante y el que contiene toda esa información tengan un mismo idioma por el cual puedan comunicarse. Los servidores web más conocidos actualmente son: Apache web server, Nginx y Microsoft.

DNS

Son las siglas de Domain Name Server, aunque la 'S' también puede hacer referencia a service o space sin necesidad de que cambie el significado ni la función del servidor, el dominio es aquel nombre que recibe una página o sitio web. El servidor DNS se encarga de relacionar una determinada dirección de dominio (www.ejemplodedominio.es) con su dirección I.P correspondiente (00.000.000.00). El funcionamiento del proceso de enlace

con la página web es el siguiente. Primeramente el usuario ingresa la dirección de dominio en su buscador, esta petición llega a un servidor DNS y el mismo transforma este dominio en su dirección IP, esta dirección IP se reconoce por el servidor web y este último proporciona el contenido de la web al usuario. El dominio se registra con el final de la dirección web (.es, .com, nl, .net, .org, ...).

Proxy server

Es un servidor de puerta de enlace. Es el encargado de conectar la red del cliente (navegador web o aplicación) al sistema externo para que se pueda realizar la solicitud de conexión mejor rendimiento y mejor accesibilidad. En otras palabras la conexión se beneficia del caché que almacena lo que acelera la carga de las páginas que el usuario ha visitado y reduce el ancho de banda de la red. Por ejemplo cuando normalmente se envía un tipo de factura a determinados destinatarios el sistema completa automáticamente uno o más campos con los datos de estos destinatarios recurrentes.

Servidor de correo electrónico

Es compatible con el flujo de correo del usuario lo que permite almacenar enviar recibir y reenviar correos electrónicos.

Servidor FTP

El nombre proviene de File Transfer Protocol también conocido como File Transfer Protocol en español. Se utiliza para transferir archivos entre el cliente y el servidor. De esta forma el servidor puede recibir el archivo del cliente y permitirle cargar el archivo al cliente. Para mejorar la velocidad se utiliza SFTP que también proporciona un servicio de seguridad encriptado. [7]

2.3 Frameworks

En el mundo del desarrollo web se utilizan muchos términos que a veces no te das cuenta. Tiene sentido que sientas una pérdida. Uno de los términos más utilizados es framework. ¿Qué es exactamente este concepto?

Un framework se puede definir como el espacio de trabajo disponible para los programadores web. Sin embargo este término cubre muchos otros conceptos por lo que a menudo causa mucha confusión a los desarrolladores web.

Cuando comenzamos con un proyecto web debemos tener una estructura ordenada y avanzada. Sin incoherencias y toda la información de un vistazo. Esta es la única forma de poder desarrollar un trabajo de calidad. En este sentido un framework es una especie de modelo diagrama o estructura conceptual basada en tecnología que nos permite trabajar de una forma mucho más sencilla. Evitando así posibles errores de programación. [8]

Si desea trabajar de una manera mucho más organizada y reducir el riesgo de error necesita un marco. Gracias a esta estructura puede aumentar su productividad así como:

- Ahorrar tiempo creando tu propio espacio de trabajo, ya que el framework viene prediseñado.
- Facilita la colaboración entre programadores, estandarizando los códigos y permitiendo trabajos colaborativos.
- Disponen de herramientas y otras utilidades que han sido diseñadas especialmente para tu framework y que puedes aprovechar.
- Un framework puede hacer que el navegador se "olvide" de centrarse solo en la funcionalidad del código.
- Evita la escritura de códigos repetitivos, como, por ejemplo, el acceso a las bases de datos o la validación de formularios.
- Permite que personas no especializadas puedan realizar programaciones que de otra manera serían realmente complicadas

2.4 Programación orientada a objetos (POO)

La programación orientada a objetos se define como un paradigma de programación una forma específica de programar en la que el código se organiza en unidades llamadas clases a partir de las cuales se relacionan los objetos entre sí para lograr el objetivo de la aplicación.

La programación orientada a objetos (POO) es una forma especial de programación más cercana a la forma en que representamos las cosas en la vida real que otros tipos de programación.

En la programación orientada a objetos tenemos que aprender a pensar en cómo resolver problemas de una manera diferente a como lo hacíamos antes en la programación estructurada. Ahora tendremos que escribir nuestro programa en forma de clases objetos propiedades métodos y otras cosas que veremos rápidamente para aclarar los conceptos y darles una pequeña ase que les permita desentrañar el concepto de este tipo de programación. [8]

2.4.1 Clases en POO

Como harás entendido las clases son la parte de declaración de los objetos también se pueden definir como la parte abstracta de los objetos. Esto significa que la definición de un objeto es clase. Cuando programamos un objeto y definimos sus características y funciones en realidad estamos programando una clase. En los ejemplos anteriores en realidad estamos hablando de clases de autos o fracciones ya que estamos a solo de determinar su forma a pesar de lo anterior.

Propiedades en clases

Una propiedad o un atributo es una característica de un objeto. Cuando definimos una propiedad normalmente especificamos su nombre y tipo. Proablemente estemos familiarizados con la idea de que las propiedades son algo así como variables donde almacenamos datos relacionados con objetos.

Métodos en las clases

Son funciones asociadas a objetos. Cuando programamos clases las llamamos métodos. Los métodos son como funciones asociadas con un objeto.

2.4.2 Objetos en POO

Un objeto es una instancia de cualquier clase. Cuando creamos una instancia debemos especificar la clase en la que se creará. Este acto de crear el objeto a partir de una clase se denomina inicialización.

2.4.3 Lenguajes en POO

Para considerar que un lenguaje de programación está orientado a objetos, se deben cumplir varias condiciones. Debe soportar objetos, debe tener un tipo o clase asociado, y

estas clases pueden heredar propiedades de una superclase o superior. (Open webinars, 2019)

Actualmente existen muchos lenguajes de programación orientados a objetos que podemos destacar: C++, Objective C (es el principal lenguaje de programación para Mac OS X iOS) Java Ruby Visual Basic Visual C Sharp Perl Smalltalk PHP o Python.

C++ junto con Java son los lenguajes de programación orientados a objetos más utilizados (según datos proporcionados por el Índice de Programación Tiobe). Por lo tanto cuando quieras acercarte a un curso de programación orientada a objetos siempre debes tener en cuenta ambos lenguajes especialmente para ampliar tu potencial de búsqueda de empleo en el mercado laboral. Python, PHP y Ruby son lenguajes de programación orientados a objetos muy populares. Popular y enfocada a la programación y desarrollo web (este tipo de programación, así como Porque la aplicación móvil es un ejemplo de programación orientada a objetos).

Hay otros lenguajes de programación más específicos como ADA, en los que te centras La seguridad se utiliza para aplicaciones de defensa. Aerolíneas, etc.

3 Metodología

3.1 Plataformas académicas digitales durante la pandemia de la COVID-19

Antes de la pandemia diferentes academias e institutos académicos manejaban metodologías de enseñanza conformadas por la convencional jerarquía del sistema educativo, manteniendo como pilar el contacto físico entre los estudiantes y los educadores, claramente es lo más recomendado a la hora de fomentar la educación, sobre todo en la enseñanza de idiomas, sin embargo, se buscaba la implementación de plataformas digitales ya sea para el seguimiento de los estudiantes, asignación de actividades para realizar por fuera de las aulas y muchas más funcionalidades que requerían un entorno virtual pero que simplemente era un modelo educativo secundario. Con la llegada de la pandemia, esto cambió considerablemente, modificar el modelo educativo principal no fue fácil, implementar un tipo de enseñanza en el que se improvisaba con la tecnología requería de un gran esfuerzo y dedicación, todo esto con el fin de continuar con la educación aunque se necesitaran otros factores metodológicos.

Durante la pandemia se realizaron diversas investigaciones, con el fin obtener hallazgos en lo que el tema respecta, principalmente en la enseñanza del idioma inglés existen resultados de entrevistas tanto a docentes como estudiantes, son análisis que se realizan luego de hacerle frente a la pandemia de manera objetiva para estudiar los resultados de las diferentes soluciones que se le dieron a los problemas e inconvenientes que ocasionó la pandemia.

Un artículo presentado por la Universidad Autónoma del Estado de Hidalgo en México indicó que dentro de todas las dificultades que se presentaron durante la pandemia a la hora de continuar con las actividades académicas, hubieron algunas que principalmente se vieron reflejadas en el decrecimiento del proceso educativo. Primeramente la existencia de la general brecha digital que se encuentra en muchos países de latinoamérica, principalmente para el acceso a internet o acceso a dispositivos tecnológicos especiales, por otro lado, la poca experiencia en el uso de las TIC por parte de los docentes e incluso en estudiantes; estos factores ocasionaron un considerable descenso en las habilidades lingüísticas que venían desarrollando los estudiantes. Claramente se le añaden los problemas que ya se presentaban previamente a la pandemia, las metodologías no eran las adecuadas y mucho menos los materiales que se utilizaban.[16]

Una de las tantas problemáticas estudiadas fueron el uso de herramientas y recursos que eran diferentes a los que se utilizaban en el proceso de enseñanza, claramente hace referencia al uso de nuevas plataformas para los estudiantes y docentes, dentro de las estrategias utilizadas como solución a esto fue el ajuste en los materiales que se solían usar y la incorporación de nuevos recursos digitales. [16]

Está claro que la pandemia trasladó a los estudiantes de un ámbito presencial a uno virtual, muy diferente a lo que estaban acostumbrados, obteniendo dificultades en todos los aspectos, cabe decir que también ha generado ventajas, por lo tanto según estudios, los estudiantes han tenido niveles de satisfacción interesantes.

Un trabajo realizado en la Escuela Superior Politécnica de Chimborazo tuvo resultados por parte de 124 estudiantes, los cuales fueron seleccionados principalmente en 3er y 4to nivel de inglés; el objetivo principal fue evaluar los niveles de satisfacción de los estudiantes en un ambiente virtual y muy diferente al que venían trabajando, el periodo de análisis fue realizado en un tiempo de 6 meses aproximadamente. La encuesta constó de 9 preguntas y su primordial objetivo era realizar un análisis estadístico y que fuese a su vez descriptivo, recolectando la información mediante preguntas abiertas. [17]

La primera pregunta fue la siguiente: ¿Qué tan satisfecho está con la disponibilidad de material de aprendizaje (libros, blogs, audios, videos)?

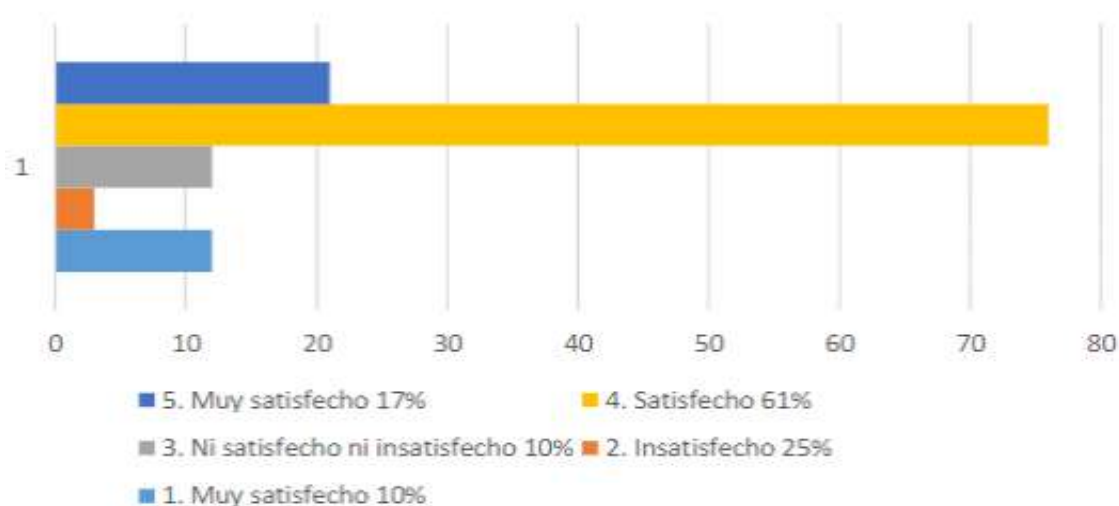


Figura 3.1: Resultados de la encuesta para la pregunta 1.

Fuente: Encuesta divulgada por Microsoft Form (Cabezas et al., 2021)

La segunda pregunta fue la siguiente: ¿Qué tan satisfecho está con la dinámica de aprendizaje-enseñanza (plataformas de enseñanza eje. Microsoft Teams, aula virtual Moodle)?

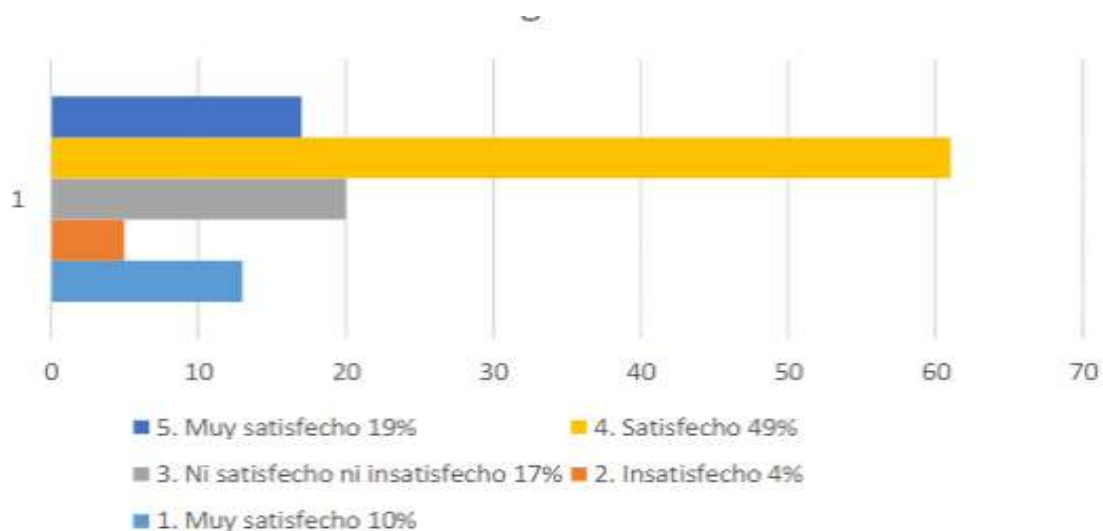


Figura 3.2: Resultados de la encuesta para la pregunta 2.

Fuente: Encuesta divulgada por Microsoft Form (Cabezas et al., 2021)

La tercera pregunta fue la siguiente: ¿Qué tan satisfecho está con la comunicación entre docente y estudiantes?

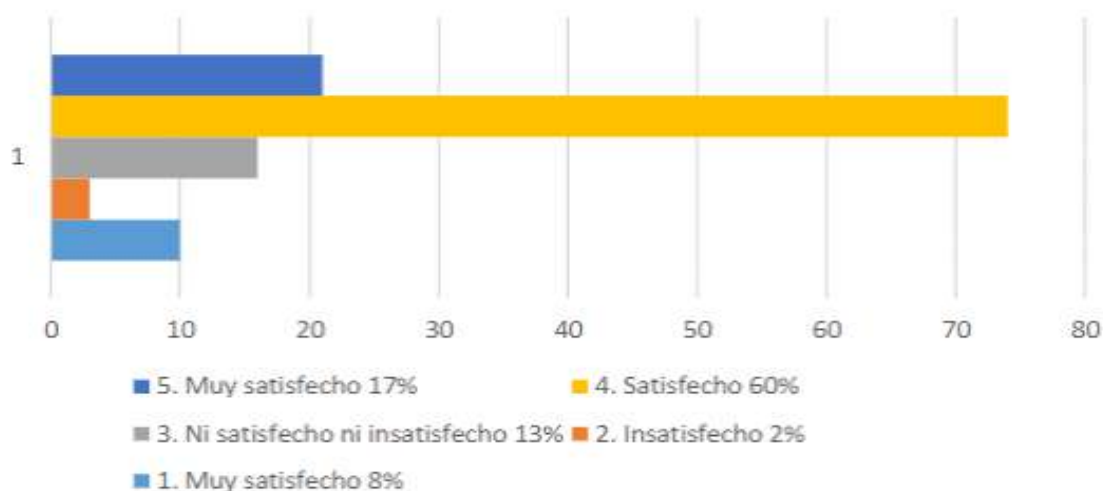


Figura 3.3: Resultados de la encuesta para la pregunta 3.

Fuente: Encuesta divulgada por Microsoft Form (Cabezas et al., 2021)

La cuarta pregunta fue la siguiente: ¿Cuán satisfecho está con la disponibilidad de la información brindada por el docente (horarios de clase, tareas, etc.)?

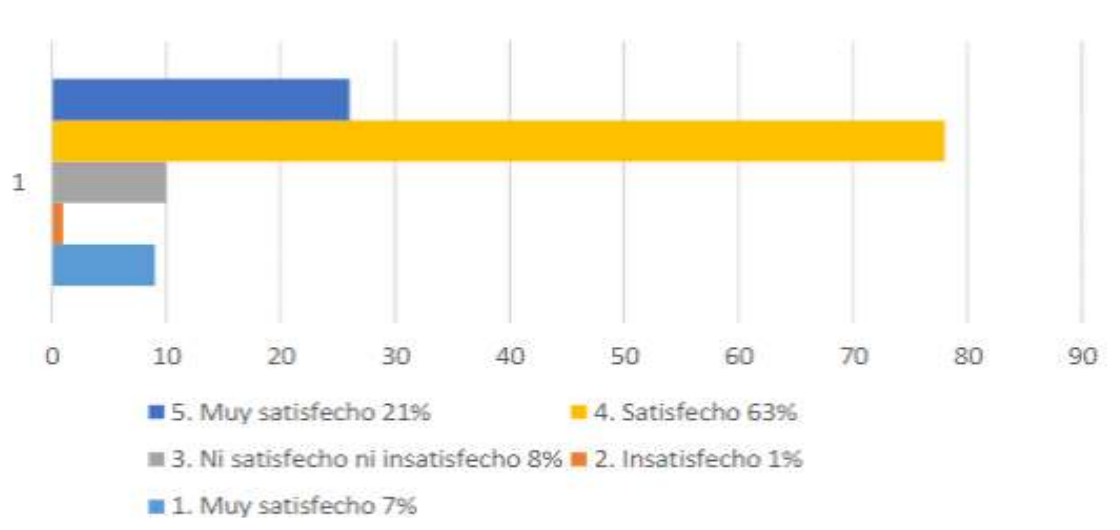


Figura 3.4: Resultados de la encuesta para la pregunta 4.

Fuente: Encuesta divulgada por Microsoft Form (Cabezas et al., 2021)

La quinta pregunta fue la siguiente: ¿Cuán satisfecho está con los métodos de enseñanza utilizados (principios y prácticas que el docente aplica en el entorno digital)?

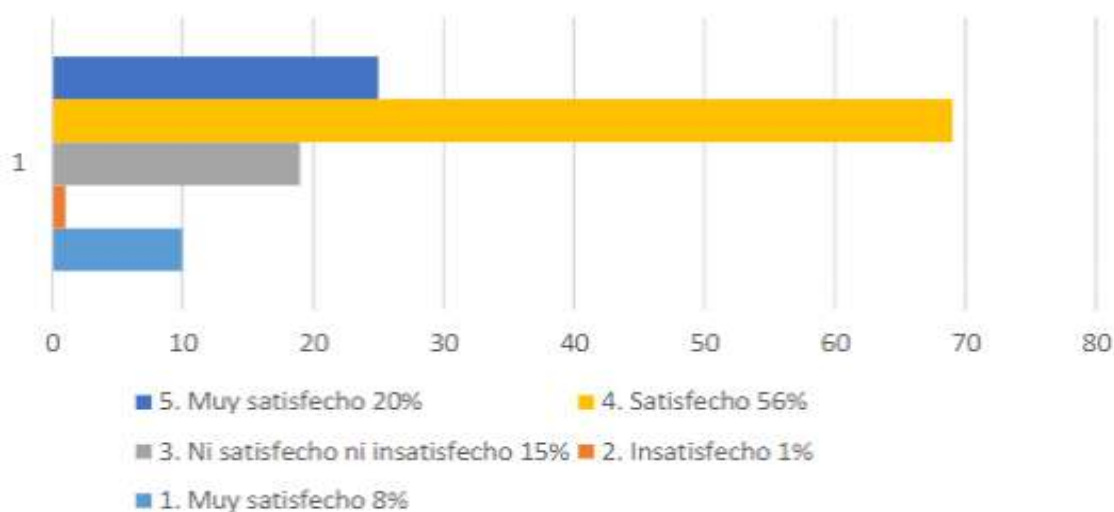


Figura 3.5: Resultados de la encuesta para la pregunta 5.

Fuente: Encuesta divulgada por Microsoft Form (Cabezas et al., 2021)

La sexta pregunta fue la siguiente: ¿Qué tan satisfecho está con la retroalimentación del docente (en clase y en relación con las pruebas)?

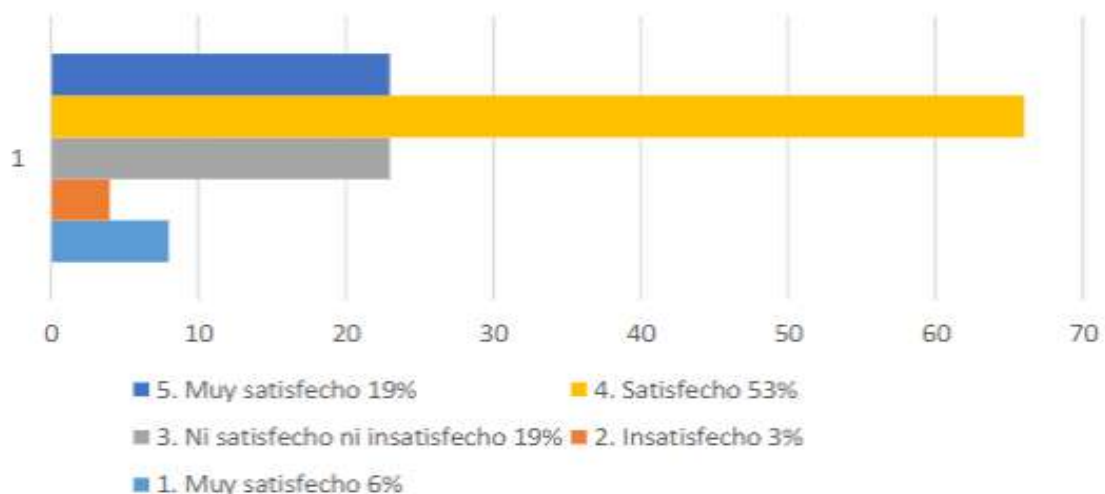


Figura 3.6: Resultados de la encuesta para la pregunta 6.

Fuente: Encuesta divulgada por Microsoft Form (Cabezas et al., 2021)

La séptima pregunta fue la siguiente: ¿Qué tan satisfecho está con la organización y aplicación de pruebas?

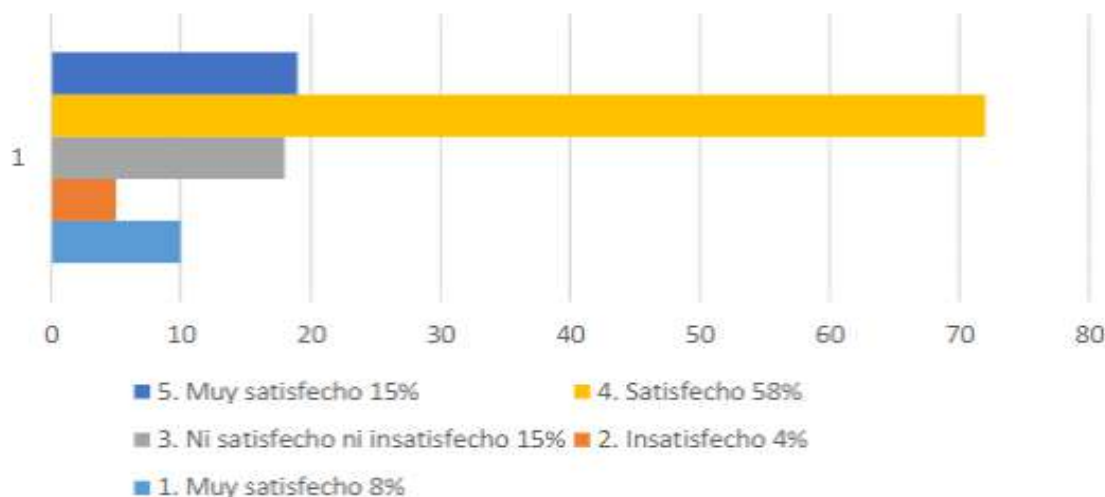


Figura 3.7: Resultados de la encuesta para la pregunta 7.

Fuente: Encuesta divulgada por Microsoft Form (Cabezas et al., 2021)

La octava pregunta fue la siguiente: ¿Qué tan satisfecho está con su desarrollo personal (académico y relacionado con el idioma)?

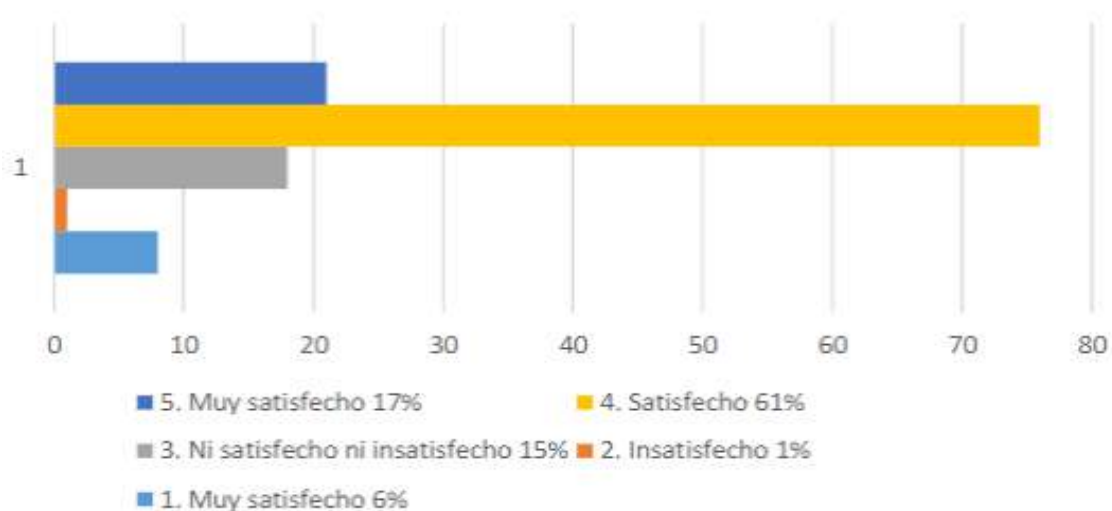


Figura 3.8: Resultados de la encuesta para la pregunta 8.

Fuente: Encuesta divulgada por Microsoft Form (Cabezas et al., 2021)

La novena pregunta fue la siguiente: ¿Cómo describiría su experiencia general relacionada con la enseñanza y el aprendizaje en un entorno totalmente digital comparado con un entorno presencial?

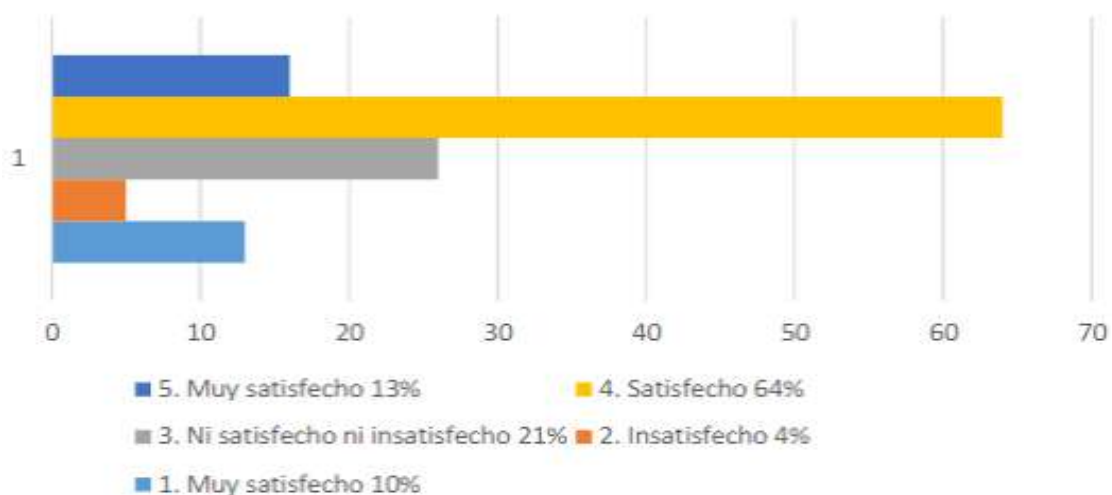


Figura 3.9: Resultados de la encuesta para la pregunta 9.

Fuente: Encuesta divulgada por Microsoft Form (Cabezas et al., 2021)

Al igual que muchos institutos, la Academia Inglés Para Todos también se vio afectada por la pandemia de la covid-19, una de las grandes desventajas es que la academia previamente dependía de una plataforma aislada a ellos, la pandemia los acarrió a una plataforma digital popular, desde entonces han usado Moodle como estrategia didáctica para continuar con la enseñanza del idioma Inglés, en paralelo y para no perder el contacto en tiempo real entre el docente y el estudiante se implementó el uso de la aplicación Zoom Meetings para videoconferencias, a medida que pasó el tiempo se fue necesitando más y más tecnología digital.

Surgió un nuevo departamento en la empresa, el cual pudiera gestionar y solucionar los inconvenientes que se presentaron, así mismo adentrar a la empresa en un mundo de aprendizaje virtual de calidad. El departamento de sistemas inició con pocos integrantes, entre ellos el practicante, la idea era desarrollar diferentes proyectos propios de la empresa, garantizando una solución digital con una imagen propia y de mucho valor. Los proyectos requeridos eran más de uno, y la exigencia por parte de los product owners era crucial para determinar el resultado de su evaluación y continuación de los integrantes, finalmente el resultado de su trabajo fue gratificante y determinante para avalar el resto del proceso.

3.2 Formación introductoria

En esta sección se describe el procedimiento inicial para la realización general de todos los proyectos, así mismo la respectiva capacitación de los ya existentes, adicionalmente se desarrolla la etapa introductoria del trabajo hecho a hasta ese momento. Las capacitaciones permiten un óptimo desarrollo de tareas que componen actividades grandes, tales como proyectos y compromisos extensos de la empresa, por ello, es fundamental esta ruta de aprendizaje.

3.2.1 Capacitación inicial

Existen diferentes proyectos hechos y requeridos en la empresa, la capacitación inicial es el primer paso para entender la magnitud de los proyectos y entornos de desarrollo que se cuentan en la academia, la exposición de las tecnologías utilizadas es sumamente importante para darle el enfoque necesario al practicante, teniendo en cuenta su nivel de aprendiz, por ello es necesario ilustrar de manera sencilla la composición general de todas las plataformas digitales y servicios ofrecidos por el departamento de sistemas de la academia. La capacitación inicial se realiza en compañía de todos los integrantes del departamento tecnológico, las personas que llevan a cabo funciones específicas y determinantes a la hora de incorporar nuevos proyectos y mejoras a los servicios digitales tanto ofrecidos como requeridos por la academia Inglés Para Todos, el avistamiento de trabajo es extenso y tiene jerarquías que lo conforman, pautando así un nivel de importancia entre proyectos y actividades durante el periodo de prácticas. El practicante deberá tomar cargo y parte de lideración en algunos proyectos, evaluando así el potencial y capacidades conformadas por el integrante del departamento.

Reforzando la capacitación se realiza la exposición de metodologías a la hora de desarrollo con equipos de trabajo, reduciendo el riesgo de falta organizacional se implementan técnicas de codificación rápidas y buenas prácticas de desarrollo de software, clasificando así tareas y manejo de repositorios utilizados por los integrantes del equipo, adicionalmente es fundamental el buen uso de estas prácticas más allá de su simple utilidad, ya que así se obtienen resultados eficientes y satisfactorios al momento de unificación y marcha de producción.

3.2.2 Información de las plataformas

Plataformas	Lenguaje utilizado	Framework de desarrollo	Utilidad
Plataforma EKT	JavaScript y PHP	Angular y Laravel	Sitio web principal de las pruebas EKT presentadas por los estudiantes de la Universidad Francisco de Paula Santander. Plataforma encargada de la inscripción de pruebas, monitoreo y resultados guardados en base de datos.
Placement Test	JavaScript y PHP	Angular y Laravel	Plataforma para la realización de pruebas de competencia inglés, es utilizada para la realización de exámenes generales en diferentes colegios, universidades e incluso personas particulares, los resultados como el puntaje y nivel de inglés dentro del marco común europeo son registrados en una base de datos aislada de los diferentes microservicios.
Plataforma IPT	JavaScript y PHP	Angular y Laravel	Es la plataforma principal de la empresa, la cual es utilizada por todos los departamentos de la academia para realizar sus determinadas funciones. Cada dependencia tiene acceso con credenciales al uso de las funcionalidades generales de la academia.
IPT World	Dart	Flutter	Proyecto aplicativo a los estudiantes registrados en la academia, cuenta con objetivos educativos y de entretenimiento. Es una aplicación móvil multiplataforma orientada al aprendizaje de inglés en diferentes niveles y modalidades de juego.

Tabla 3.1: Plataformas y servicios digitales de la academia Inglés Para Todos

Fuente: Autor.

3.3 Ruta de aprendizaje

Se da inicio a una ruta de aprendizaje introductoria y preparativa para el practicante, la cual es prolongada y evaluada continuamente permitiendo una etapa de capacitación primordial para el desarrollo general de todos los proyectos posibles. Existen diferentes plataformas educativas para tomar este tipo de aprendizaje de áreas en desarrollo de software, las cuales son muy utilizadas en la industria tecnológica; hoy en día el aprendizaje online permite a las personas capacitarse en áreas que deseen, obteniendo así conocimientos y aptitudes significativas dentro del ámbito laboral, profesional y personal.

Platzi es una de las herramientas más utilizadas en la educación online, la ruta de aprendizaje definida se basa principalmente en esa institución, tanto la realización de cursos como las diferentes capacitaciones dentro de la empresa son de muchísimo valor para el aprendiz. Por otro lado existen diferentes herramientas gratuitas para complementar ese conocimiento, tales como documentación, videos y en este caso, la realización de cursos en la plataforma Udemy, los cuales se realizan más adelante.

La ruta de aprendizaje definida se divide en tres secciones, cada sección se complementa una después de la otra y se definen tiempos para la terminación de cada una, adicionalmente se realizan presentaciones individuales de cada curso con el fin de evaluar los conocimientos adquiridos.

3.3.1 Primera sección de aprendizaje

Esta primera sección conforma los cursos introductorios al desarrollo de software, los cuales son la base de todo para adquirir habilidades en diferentes tecnologías, se podría decir que es la etapa inicial del aprendiz para convertirse en un desarrollador Full-Stack Junior.

Curso profesional de Git y GitHub

Cuando hablamos de un control de versiones nos referimos a todos esos sistemas que nos permiten monitorear en tiempo real los cambios realizados en un proyecto por el equipo de trabajo, ya sea que esté conformado por una o varias personas. Hoy en día es ideal mencionar que Git es el control de versiones más moderno que existe, ya que por su versatilidad es el más usado en el mundo. [9]

GitHub es una de las plataformas que nos facilitan la creación de proyectos que se encuentren abiertos para su uso mediante herramientas y aplicaciones, GitHub es prácticamente una red social para programadores, ya que es un portal que nos permite alojar todo tipo de aplicaciones y códigos, permite realizar colaboraciones con otros desarrolladores, crear ramificaciones a partir del proyecto inicial y muchas tareas más que se hacen interesantes para un programador. [10]

La realización del curso de Git y GitHub como primero de muchos, da acceso a un mundo tecnológico y su principal funcionamiento, tener una organización con los repositorios de los proyectos es ideal para el trabajo, se logran planificaciones eficientes y muy profesionales. Con 6 horas de contenido se llevan a cabo 16 horas de práctica, adicionando a eso 16 horas más de planificación y evaluación.

Dentro de los propósitos de este sistema de versiones y su portal encontramos el monitoreo de registros, llevar registros es fundamental, principalmente llevarlos de manera detallada respecto a los cambios realizados a lo largo del desarrollo general; la coordinación es otro propósito que permite un flujo organizado del trabajo llevado a cabo por múltiples personas; el control también es un propósito que nos permite guardar solamente los cambios realizado en algún archivo o en su defecto, en el proyecto general, finalmente la claridad es necesaria, dejar claro donde ocurrieron los cambios, cuando ocurrieron y quien o quienes hicieron dichos cambios.

Comando	Función
git init	Lo usamos para determinar la carpeta en la que vamos a trabajar.
git status	lo usamos para saber si tenemos un archivo añadido o borrado en nuestro proyecto, para saber en la rama en la que estamos y si tenemos commits.
git add .	Es para añadir un archivo a nuestra rama seguidamente ponemos entre comillas el nombre de nuestro archivo o poner un punto para añadir todos los archivos de nuestra carpeta.
git rm	Lo usamos para borrar un archivo que hayamos añadido, para eliminarlo por completo de nuestra rama usamos <code>git rm -cached</code> .
git commit -m	Se usa para añadir un commit a nuestra rama, también podemos ponerle un -m seguidamente ponemos entre comillas nuestro mensaje.
git config	Muestra configuraciones de git también podemos usar <code>-list</code> para mostrar la configuración por defecto de nuestro git y si añadimos <code>-show-origin</code> inhales nos muestra las configuraciones guardadas y su ubicación.

Tabla 3.2: Comandos principales utilizados en Git

Fuente: Autor.

Introducción a la terminal y línea de comandos

El curso de introducción a la terminal y línea de comandos es partir desde lo más básico del mundo de la programación, son herramientas pilares para un desarrollador. Aprender a utilizar la terminal nos permite ser mucho más flexibles a la hora de codificar, en complementación a eso, permite volvernós rápidos al momento de ejecutar tareas ya que no se cuenta con una interfaz gráfica que disminuya el flujo de procesos.

La terminal viene siendo esa interfaz gráfica que nos permite visualizar una línea de comandos, y cuando hablamos de líneas de comandos nos referimos a un shell, en pocas palabras, la terminal es una ventana que nos muestra el prompt, el prompt es aquel que nos indica mediante conjunto de caracteres, la espera de una orden. La línea de comandos vienen siendo las series de instrucciones que se pasan al sistema operativo para realizar una tarea en específico.

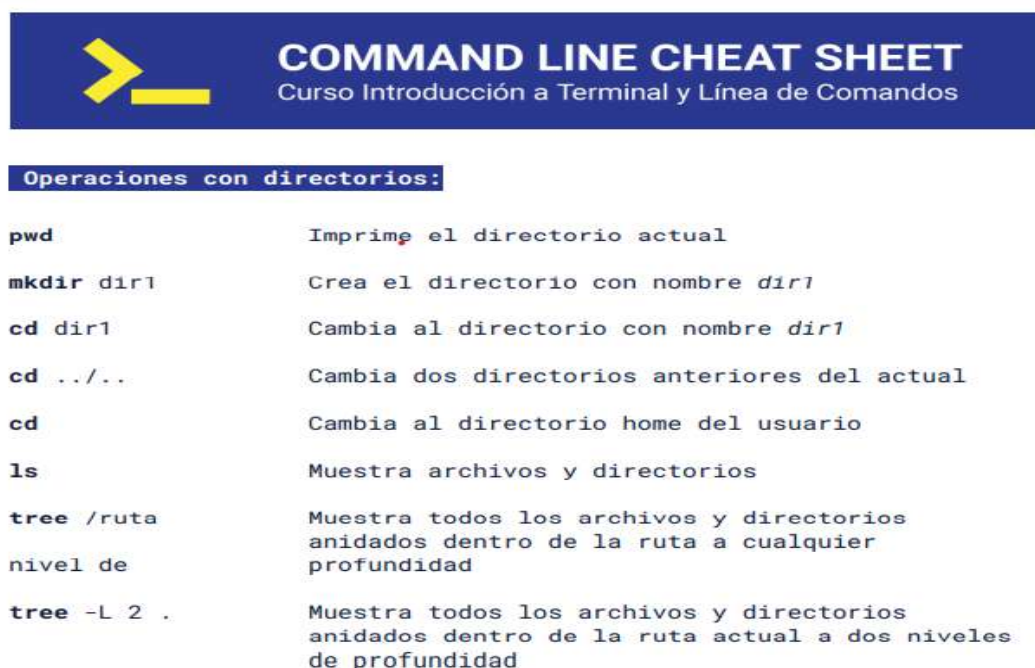


Figura 3.10: Comandos básicos.

Fuente: Platzi

Algoritmos y pensamiento lógico

El desarrollo de pensamiento lógico es uno de los pilares para entender el mundo del desarrollo web, de hecho es fundamental para entender todo el mundo de la programación, así mismo, contribuye a convertirse en desarrollador de software. Los conceptos de programación, variables, tipos de datos, sistema binario, ciclos, estructuras de control y condicionales son pautados para dar inicio al entendimiento y desarrollo como profesional, este curso permite el flujo de conocimientos de diseño de algoritmos y transformación en códigos de programación para la industria de la tecnología.

Conocer el sistema binario es poder entender el funcionamiento principal de una computadora, diariamente conocemos el sistema decimal, pero el binario es el que entiende la computadora, puede representar 1 y 0, alto y bajo, encendido y apagado, positivo y negativo, true y false, entre otros. Básicamente el curso de algoritmos y pensamiento lógico es la complementación de las asignaturas de programación dictadas en ingeniería, ya que se refuerzan ciertos temas como variables y constantes, tipos de datos (Enteros, texto, lógicos), operacionales, composición, arrays, tablas de verdad, estructuras de control y muchos subtemas más.

Buenas prácticas para escritura de código

Para evitar los errores más comunes en las escrituras de código se necesitan de buenas prácticas a la hora de realizarlos, ese es el objetivo principal de este curso, poder aprender reglas y normativas que ayuden a generar una buena tendencia a la correcta escritura de códigos.

Tener un código bien escrito beneficia principalmente al autor, generalmente cuando es un proyecto que se retoma luego de mucho tiempo; luego beneficia a los compañeros de trabajo, a los integrantes del proyecto o a los futuros contribuyentes del mismo ya que a la hora de realizar modificaciones o actualizaciones el simple hecho de que otros desarrolladores entiendan la lógica del código hace que estén bien aplicadas esas buenas prácticas de escritura. Finalmente beneficia al cliente, aunque no tenga los mismos conocimientos que los desarrolladores, es importante que entienda el proyecto de manera superficial, así su lógica de negocio estará mejor atendida.

Características internas de escrituras de código

- **Legibilidad:** Que tan fácil es poder entender e interpretar lo que el código dice, en pocas palabras es cuánto esfuerzo implica entender lo que el código dice sin ejecutarlo, un factor que puede afectar la legibilidad es la falta de indentación.
 - **Mantenibilidad:** Tiene que ver mucho con el futuro, la comparación entre el proyecto o software que escribimos hoy con el de más adelante, es decir, cuánto esfuerzo supondrá adaptar el código a nuevos requerimientos.
-

- Testeabilidad: Cuánto esfuerzo nos lleva realizar pruebas para demostrar que el código es funcional.

Código prolijo

A screenshot of a code editor showing PHP code with poor indentation. The code is a script that builds a SQL query based on form input. It uses multiple if-statements to check for the presence of 'txtFechaInicial', 'txtFechaFinal', 'txtTrabajador', and 'txtSerie' in the \$_POST array. Each if-statement sets a corresponding part of the \$where array. The final \$query is constructed using implode. The code is 21 lines long and lacks consistent indentation for its conditional blocks.

```
1 <?php
2 if(isset($_POST['txtFechaInicial']) && $_POST['txtFechaInicial']!="")
3 {
4     $where[] = "fecha_recibo > " . $_POST['txtFechaInicial'] . "";
5 }
6 if(isset($_POST['txtFechaFinal']) && $_POST['txtFechaFinal']!="")
7 {
8     $where[] = "fecha_recibo < " . $_POST['txtFechaFinal'] . "";
9 }
10 if(isset($_POST['txtTrabajador']) && $_POST['txtTrabajador']!="")
11 {
12     $where[] = "trab_recibo = " . $_POST['txtTrabajador'] . "";
13 }
14 if(isset($_POST['txtSerie']) && $_POST['txtSerie']!="")
15 {
16     $where[] = "serie1 = " . $_POST['txtSerie'] . "";
17 }
18
19 $query = "SELECT * FROM recibo WHERE '" . implode(" AND ", $where) . "'";
20 $result = mysqli_query($conexion, $query) or die(mysqli_error($conexion));
21 ?>
```

Figura 3.11: Código con mala indentación.

Fuente: Platzi

A screenshot of a code editor showing the same PHP code as Figure 3.11, but with good indentation. The code is 17 lines long. The conditional blocks are properly indented using curly braces, making the structure of the code much clearer and easier to read.

```
1 <?php
2     if (isset($_POST['txtFechaInicial']) && $_POST['txtFechaInicial'] != "") {
3         $where[] = "fecha_recibo > " . $_POST['txtFechaInicial'] . "";
4     }
5     if (isset($_POST['txtFechaFinal']) && $_POST['txtFechaFinal'] != "") {
6         $where[] = "fecha_recibo < " . $_POST['txtFechaFinal'] . "";
7     }
8     if (isset($_POST['txtTrabajador']) && $_POST['txtTrabajador'] != "") {
9         $where[] = "trab_recibo = " . $_POST['txtTrabajador'] . "";
10    }
11    if (isset($_POST['txtSerie']) && $_POST['txtSerie'] != "") {
12        $where[] = "serie1 = " . $_POST['txtSerie'] . "";
13    }
14
15    $query = "SELECT * FROM recibo WHERE '" . implode(" AND ", $where) . "'";
16    $result = mysqli_query($conexion, $query) or die(mysqli_error($conexion));
17    ?>
```

Figura 3.12: Código con buena indentación.

Fuente: Platzi

Programación orientada a objetos POO

La programación orientada a objetos es clave para el desarrollo de software, principalmente el desarrollo de aplicaciones web, aplicaciones móviles, sitios web y todo tipo de proyectos. Nos permite programar más rápido y tener un mejor dominio de conceptos para ejecutar proyectos.

La POO se basa en cuatro características principales:

- Encapsulamiento: se refiere a ocultar datos por medio de códigos.
- Abstracción: Es la representación de un objeto mediante código.
- Herencia: Características que tiene una clase nueva a partir de una existente.
- Polimorfismo: Se refiere a la propiedad por la que es posible enviar mensajes sintácticamente iguales a objetos de tipos distintos.

La programación orientada a objetos nace de los problemas que la programación estructurada nos dejó, como la tendencia a un código muy largo, situaciones en las que si algo falla todo se rompe, códigos difíciles de mantener con el tiempo, código spaghetti (if, else, if else...), etc. La POO nos permite observar los problemas en forma de objetos. Dentro de los paradigmas de la programación orientada a objetos encontramos:

- Clases: Una clase es un modelo por el cual se va a construir determinado objeto, cada clase permite generar muchos más objetos.
 - Propiedades: Las propiedades también se pueden llamar atributos, como por ejemplo si queremos instanciar una clase que viene siendo un perro, sus propiedades pueden ser su nombre, edad, peso, color, raza, altura y sexo.
 - Métodos: Como su nombre lo dice, los métodos nos permiten definir lo que un objeto puede hacer por medio de su algoritmo.
 - Objetos: Es la instancia de una clase.
 - Encapsulamiento: El encapsulamiento es plantear y definir un dato para que no sea modificable, alterable o violable, esto siempre y cuando se le asigne un modificador de acceso, específicamente para que no solo el dato se oculte si no que se protega.
 - Abstracción: Nos permite extraer los parámetros más importantes o relevantes de una clase, en pocas palabras es separar la información central de los detalles secundarios.
-

- Herencia: Como su nombre bien lo indica, nos permite crear nuevas clases a partir de otras, esto generalmente se realiza cuando se detectan características y comportamientos iguales en una clase.
- Polimorfismo: Contruir métodos con el mismo nombre pero con comportamientos diferentes.

Lenguajes de programación orientados a objetos



Figura 3.13: Lenguajes de programación orientados a objetos.

Fuente: Platzi



Figura 3.14: Lenguajes de programación.

Fuente: Platzi

3.3.2 Segunda sección de aprendizaje

En esta sección de aprendizaje se centralizan tecnologías utilizadas en desarrollo backend, primordial para la funcionalidad de proyectos y mantenimiento de los mismos. Aptitudes para dominar serían la interconexión de servicios, peticiones http con API's, manejo de bases de datos y usos de framework para lenguajes de programación con finalidades de desarrollo backend.

Introducción a PHP con Composer

PHP es un lenguaje de programación que se puede adaptar fácilmente al desarrollo web del lado del servidor, por lo tanto la realización de este curso permite al aprendiz poder conectar los servicios y gran parte del desarrollo backend a la lógica de negocio requerida por la empresa.

Es un lenguaje muy flexible y fácil de aprender, tiene gran documentación en español lo cual lo vuelve una ventaja, con ello se caracteriza por ser bastante demandado. Podemos encontrar Frameworks populares como Laravel y un poco de Wordpress, y lo más importante es que gran parte de los sitios web lo usan.

Composer no es más que un conjunto de herramientas sistematizadas para gestionar paquetes que permitan la programación en PHP, así mismo provee varios estándares de formatos que son necesarios para la manipulación de librerías y todo tipo de dependencia.

- Cliente y servidor: Nuestra aplicación permanece guardada completamente en un servidor, muchas empresas ofrecen tal servicio para que diversos sitios web sean alojados en dichos servidores, los cuales entregan una copia de la misma a cada cliente que lo solicite.
 - Dominio: El dominio viene siendo esa dirección principal o título de nuestro enlace, es nuestra identificación en internet y gracias a ella cualquier computador es capaz de encontrar páginas web.
 - Servidor físico o VPS: Es la que se encarga de mantener guardada y disponible las 24 horas la página web, también se le puede llamar servidor y casi siempre está conectado a internet, incluso a partir de él podemos definir y asignar reglas de seguridad al momento de navegar.
 - Servidor web: En cuanto al servidor web, este viene siendo una especie de programa que corre dentro del servidor físico (VPS), se encarga de la gestión de cualquier petición que se realice.
 - Métodos HTTP: Los métodos HTTP son aquellos que nos permiten formar una comunicación entre el navegador y el cliente, funcionan con una solicitud que
-

realiza el cliente en la cual le pide al servidor realizar una acción en específico o múltiples acciones.

- GET: Este método es de los más populares usados en el desarrollo web de lado del servidor ya que nos permite solicitar información al servidor, por ejemplo podemos solicitar un listado de diferentes productos en caso de que estemos haciendo una compra o un listado de cursos en caso tal estemos solicitándolos en una academia virtual.
- POST: El método post nos permite guardar información, como ejemplo de aplicación, este método nos facilita la recolección de datos de un usuario ingresados en un formulario para luego llevarlos a una base de datos en el servidor donde serán procesados.
- PUT/PATCH: Por otro lado, este par de métodos nos permiten actualizar cualquier tipo de información que ya esté guardada previamente mediante un método post, se pueden editar como por ejemplo correos electrónicos, números de teléfono, etc. Existe una diferencia en ambos, y es que PUT reemplaza toda la información y PATCH solo actualiza la información necesaria
- DELETE: Como su nombre lo indica, con este método podemos eliminar un recurso o información del servidor.

Introducción a Laravel

Laravel no es más que una herramienta construida en PHP y es importante conocer este lenguaje para luego utilizar y manejar eficazmente este framework, en pocas palabras es una herramienta de desarrollo web, nos permite desarrollar páginas web, sistemas, API's, etc.

Cabe recalcar que la realización de este curso le permite al aprendiz desarrollar aplicaciones web desde cero y de manera rápida, ya que se le permite al alumno el uso de un motor para plantilla Blade, aplicando diseños de desarrollos hechos en Bootstrap y pasando su conexión de servicios mediante bases de datos usando Eloquent ORM, algo por agregar sería el aprendizaje de metodologías CRUD.

Fundamentos de bases de datos

Las bases de datos son muy importante en el ámbito de desarrollo de software ya que nos permite tener un sólido respaldo de nuestros proyectos y guardar todos los recursos necesarios. Las bases de datos surgieron de ello, de la necesidad de mantener respaldada la información sin importar las limitaciones que se nos presenten en una memoria RAM.

El aprendiz es beneficiado con la realización de este curso, debe comprender el funcionamiento de un proyecto con alta cantidad de datos y así mismo comprender el flujo

que este conlleva, es importante tener las bases de programación para ejecutar algoritmos y scripts que se necesiten para hacer tareas específicas. Se obtienen conocimientos en entidades y atributos, son propiedades clave de las bases de datos relacionales.

El manejo de MySQL también es importante, tanto su instalación como la utilización frecuente de la misma:

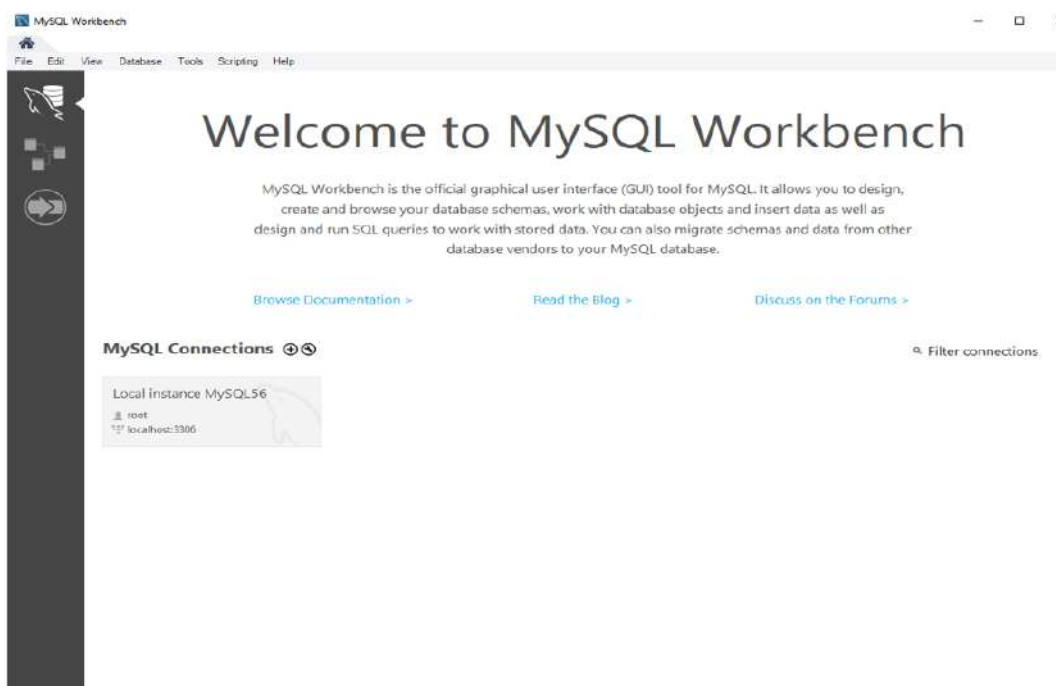


Figura 3.15: Instalación de MySQL.

Fuente: Platzi

SQL significa Structured Query Language y tiene una estructura clara y fija. Su objetivo es hacer un solo lenguaje para consultar cualquier manejador de bases de datos volviéndose un gran estándar. En él podemos encontrar gran sublenguaje, DDL o Data Definition Language que nos ayuda a crear la estructura de una base de datos. Existen 3 grandes comandos:

- Create: Nos ayuda a crear bases de datos, tablas, vistas, índices, etc.
- Alter: Ayuda a alterar o modificar entidades.
- Drop: Nos ayuda a borrar. Hay que tener cuidado al utilizarlo.

SQL práctico

Se podría decir que es el representante de lenguajes para bases de datos y manejo de datos estructurados, eficientes al momento de aplicar álgebra relacional, subconsultas

y funciones de ventanas. SQL está basado en dos principios fundamentales cuando a lo teórico se refiere, específicamente a la teoría de conjuntos la cual podemos ver representadas por ejemplo en diagramas de venn, el segundo principio vendría siendo el algebra relacional de Edgar Codd. Algo interesante es que SQL nació en los años 70 y originalmente fue llamada SEQUEL, sin embargo ese nombre fue cambiado por diversos inconvenientes presentados por otra empresa la cual contaba con ese nombre y derechos de autor del mismo. [11]

El curso práctico de SQL le permite al aprendiz reforzar los conocimientos teóricos del curso de fundamentos de bases de datos, para así aplicar de manera genérica comandos descriptivos y realizar consultas con servicios API enlazados a los proyectos requeridos por la empresa, cabe recalcar que el manejo de datos en la misma es bastante grande y requiere de un alto cuidado y concentración por parte del practicante y más adelante se verán reflejados estas prácticas dentro de los proyectos a realizar.

Manejo de datos en Laravel con Eloquent

El objetivo principal de este curso es que el aprendiz use las bases obtenidas de PHP y todo lo relacionado con bases de datos para aprender a configurar modelos de objetos relacionales y así mismo mapearlos en el framework Laravel utilizando Eloquent.

Eloquent es un ORM integrado en Laravel que permite configurar y llevar a cabo de manera rápida y eficaz procesamiento de datos de un determinado proyecto, esto es debido a la diversidad de funciones que incluye, obteniendo así un modelo eficiente para realizar consultas complejas. [12]

Con este curso el practicante aprende el concepto de un ORM, se le presenta un caso para resolver y como consecuencia debe realizar un mapeo de objetos relacionales, y esto se logra con un ORM que no es más que un mapeador de objetos relacionales, en otros términos se puede definir como un modelo de programación avanzado y es importante este curso en la ruta de aprendizaje ya que va acorde con la metodología CRUD la cual es utilizada en equipo.

Curso de API Rest

Este curso es clave para el practicante, ya que es sumamente importante el conocimiento de API's (Application Programming Interface) para la ejecución de todos los proyectos requeridos, aprender a desarrollar y configurar entornos para aplicar servicios REST es gratificante para las funcionalidades del backend de un proyecto, aplicación web o página web conectada a bases de datos. Estos servicios REST a implementar son los más comunes como GET, POST, DELETE y PUT. El aprendiz aprenderá específicamente a:

- Entender los conceptos principales de REST
-

- Aprender a consumir servicios REST
- Producir servicios REST
- Restringir el acceso a las API Rest

Una API se encarga de definir el conjunto de reglas por las cuales dos aplicaciones interactúan entre sí, es decir, que tipo de mensajes se envían, cual aplicación inicia esa comunicación, cual de las dos respalda esa información, que tipos de respuestas se deben esperar, etc.

Curso de Postman

Postman se caracteriza por ser una plataforma que se encarga de colaborar en el desarrollo y testing de APIs, es decir, de probar las APIs creadas para determinados proyectos. Así mismo Postman es una herramienta que nos permite organizar todo tipo de peticiones en formato de colecciones y automatiza la integración de los servicios API estipulados con las aplicaciones. [13]

Postman es la herramienta que se utiliza en la Academia Inglés Para Todos para hacer testing de los proyectos requeridos y los ya estabilizados, es importante el buen manejo de la misma por parte del aprendiz para testear los proyectos en los que esté trabajando junto al equipo de desarrollo. Cabe resaltar que primeramente se utilizará para el testeo de una de las plataformas a desarrollar y posteriormente en las generales

3.3.3 Tercera sección de aprendizaje

En la sección de aprendizaje final se plantea la capacitación de tecnologías para el desarrollo frontend, en el cual se implican funcionalidades de interfaz de usuario y experiencia de usuario, así como toda la parte visual de aplicaciones web y móviles, adicionalmente se le capacita al aprendiz con temas de diseño.

HTML y CSS práctico

Es importante mencionar que todo desarrollador en general debe conocer tecnologías como HTML y CSS, saber manejarlas y entenderlas ya que es la base de toda página web, sitio web o plataforma, por un lado, HTML nos permite estructurar toda la información que queremos mostrarle al usuario por medio del proyecto que se esté realizando, principalmente se trabaja con etiquetas, ya sean contenedoras o específicas que se abren y cierran en cierto punto, sin embargo, si no son de contenido, no siempre se cierran. Por otro lado, CSS nos ayuda a darle estilos a esa página web que estamos diseñando o maquetando, permite una buena presentación del proyecto en cuanto a diseños y estilos, sus herramientas se caracterizan por presentar un ';', es un selector que se encarga básicamente de la declaración de una propiedad y su valor, cabe resaltar que HTML y CSS no son lenguajes de programación como lo es JavaScript.

Es importante que el practicante tenga conocimientos previos de estas tecnologías, ya que al ser herramientas de desarrollo frontend básicas, se espera que el aprendiz tenga un manejo aceptable de las mismas, así mismo la realización de este curso le permite reforzar esas habilidades de maquetación, se selecciona este curso entre muchos por su brevedad y proyecto corto que es la realización de un clone del navegador de Google.

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <meta http-equiv="X-UA-Compatible" content="ie=edge" />
7     <title>Google Clone</title>
8     <link rel="stylesheet" href="css/main.css" />
9   </head>
10  <body>
11    <header>
12      <nav>
13        <ul class="nav-right-section">
14          <li>
15            <a href="#">Correo</a>
16          </li>
17          <li>
18            <a href="#">Imágenes</a>
19          </li>
20          <li class="menu-icon">
21            <a href="#"></a>
22          </li>
23          <li>
24            <a href="#">
25              
29            </a>
30          </li>
31        </ul>
32      </nav>
33    </header>
34    <main>
35      <section class="main-logo">
36        <img src="" alt="" />
37      </section>
38      <section class="main-input">
39        <div class="main-input-container">
40          <span class="search-icon"></span>
41          <input type="text" />
42          <a class="micro-icon" href="#"></a>
43        </div>
44      </section>
45      <section class="main-buttons">
46        <div>
47          <button>Buscar con Google</button>
48        </div>
49        <div>
50          <button>Me siento con suerte</button>
51        </div>
52      </section>
53    </main>
54  </body>
55 </html>
```

Figura 3.16: Código HTML de clone de Google.

Fuente: Platzi y Autor

```
1  body {
2    margin: 0;
3    padding: 0;
4    font-size: 13px;
5    font-family: Arial, Helvetica, sans-serif;
6  }
7  a {
8    text-decoration: none;
9    cursor: pointer;
10 }
11
12 header {
13   width: 100%;
14   height: 60px;
15 }
16 header nav {
17   display: flex;
18   justify-content: flex-end;
19 }
20 header nav .nav-right-section {
21   width: 250px;
22   height: auto;
23   display: flex;
24   list-style: none;
25   justify-content: center;
26   align-items: center;
27 }
28 nav .nav-right-section a {
29   margin-right: 10px;
30   color: #000000;
31 }
32 nav .nav-right-section .menu-icon {
33   background-image: url("https://static.thenounproject.com/png/756729-200.png");
34   background-repeat: no-repeat;
35   background-position: center;
36   background-size: contain;
37   width: 25px;
38   height: 25px;
39 }
40 nav .nav-right-section img {
41   border-radius: 50%;
42   margin-left: 10px;
43 }
44
45 main {
46   margin-top: 150px;
47   text-align: center;
48 }
49 main .main-logo {
50   width: 530px;
51   margin: 0 auto;
52   margin-bottom: 35px;
53 }
54 main .main-logo img {
55   width: 300px;
56 }
57
```

Figura 3.17: Código CSS de clone de Google.

Fuente: Platzi y Autor

Sistemas de diseño para desarrolladores

Todo desarrollador web debe tener conocimientos básicos en diseño, manejar sistemas y patrones de diseño acorde a la funcionalidad necesaria. Este curso le permitirá al aprendiz crear sistemas de diseño, dominar la instalación de un sistema de diseño, aplicar flujos de trabajo, revisar y testear componentes y muchas cosas más.

Específicamente un sistema de diseño tiene como objetivo mejorar el flujo de trabajo de un equipo permitiendo aprender a iterar mucho trabajo de manera eficiente y rápida. El desarrollador se vuelve un poco independiente al momento de aprender a diseñar por su propia cuenta, ya que no está a la espera de un miembro del equipo para depender de su tiempo y ver afectado su progreso.

Durante el curso se realizan los siguientes ítems:

- Personalización de la terminal de comandos
 - Instalación y manejo de Node y Express desde cero
 - Bonus de HTML, CSS y JS
 - Automatización de diseños
 - Pre procesadores
 - SMACSS - Arquitectura de CSS escalable
 - BEM - Bloque elemento modificador
 - Bootstrap
 - CSS Grid
 - Animaciones en CSS
-

Curso de Frontend Developer

El curso de frontend developer le permite al aprendiz reforzar de gran manera el manejo de HTML y CSS para elaborar maquetación de páginas web de manera más eficiente y profesional, adicionalmente se hace uso de una nueva tecnología para el practicante la cual es el responsive design, es una metodología que permite adaptar un sitio web a cualquier dispositivo. Para llegar acabo este patrón de desarrollo, se hace uso a la vez de otra tecnología, mobile first, la cual explica que para la elaboración y desarrollo de páginas web, se debe partir desde dispositivos más pequeños, hasta los más grandes, es decir, iniciar un diseño escalable.

Durante el curso se realizan los siguientes ítems:

- Selectores, propiedades y valores en CSS
- Selectores básicos y combinados
- Pseudoclases y pseudoelementos
- Cascada y especificidad en CSS
- Tipos de display: Block, inline e inline-block
- Flexbox y CSS Grid
- Modelo de caja
- Z-index y el contexto de apilamiento
- Unidades de medida
- Responsive design
- Atomic Design

Flutter

Flutter es un framework de código abierto muy popular, permite crear aplicaciones de manera rápida, obteniendo diseños hermosos y multiplataforma, esto quiere decir que serán soportadas para todos los sistemas operativos, son compiladas de forma nativa que parte desde una sola base de código estandarizada. [14]

Esta tecnología es la utilizada para la elaboración de proyectos orientados a aplicaciones móviles dentro de la empresa, es por ello que el practicante debe manejar muy bien este framework y su lenguaje de programación el cual es Dart.

Este curso tiene un contenido introductorio y a su vez básico-intermedio que permite adentrar al aprendiz en un ámbito más amigable con el desarrollo de aplicaciones móviles, Flutter al ser un SDK (Kit de desarrollo de software) creado por Google, contribuye al diseño de interfaces nativas iOS y/o Android, y esto es lo que se necesita para llevar a cabo los proyectos requeridos por la empresa respecto a aplicaciones móviles.

Aplicación realizada en el curso:

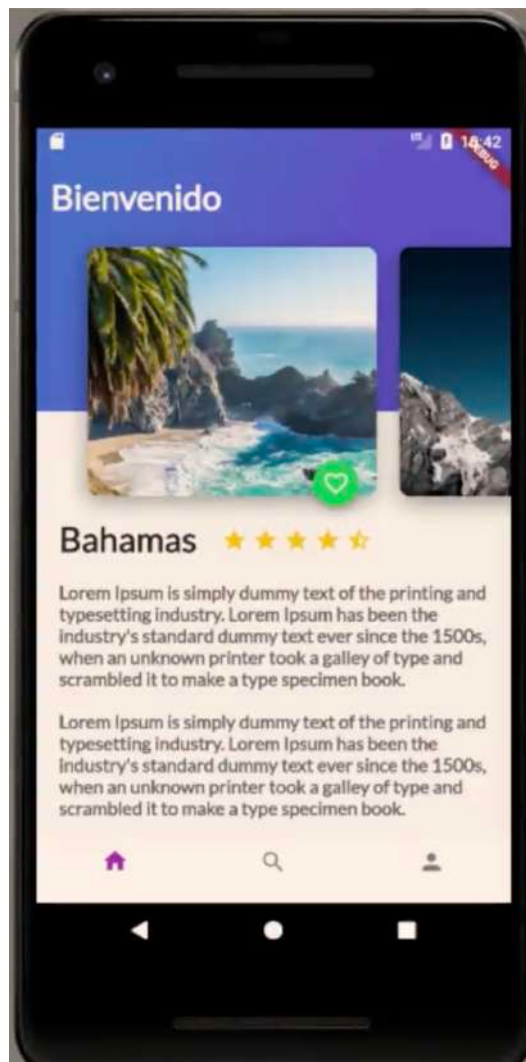


Figura 3.18: Aplicación de viajes desarrollada en Flutter.

Fuente: Platzi y Autor

Flutter avanzado

El curso de Flutter avanzando le contribuye al aprendiz una intensa profundización de lo aprendido en el curso básico de Flutter, llevando a otro nivel las metodologías de desarrollo, patrones de diseño y arquitecturas de software. Se introducen nuevos conceptos un poco más profesionales y orientados a lógicas de negocio, lo cual es lo que se necesita para el desarrollo de proyectos aplicativos por parte del practicante dentro de la empresa.

Es importante que las aplicaciones desarrolladas en Flutter tengan una arquitectura de software definida ya que eso nos permite tener un código legible y profesional frente a una comunidad de desarrolladores y de empresas en general, usar estos patrones de desarrollo nos permiten estructurar, diseñar y proyectar de mejor manera cualquier proyecto requerido. Es fundamental tener claro las dos siguientes ideas que definen las arquitecturas de software a la hora de manejar Flutter de forma más avanzada y profesional:

- Indica la estructura, funcionamiento e interacción entre las partes del software.
- Tener el código organizado para que todo el equipo pueda verlo.

Existen diferentes arquitecturas de software para Flutter, a continuación se definen las siguientes:

- Vanilla: En esta arquitectura, dentro del mismo Widget se coloca la lógica del mismo y su vista principal, el beneficio esperado es que se vuelve simple y autónoma, y que al momento de estar conectado en cualquier parte de la aplicación siempre estará recuperando datos y haciendo su respectivo renderizado. Por otro lado, así como existe un beneficio, existe una desventaja de esta arquitectura de software que afecta a largo plazo, y es que escribir Widgets de esta manera implicaría dificultad para su legibilidad siempre y cuando la lógica se vuelva más extensa y avanzada, ocasionando así bifurcaciones en el código. A pesar de que esta arquitectura es fácil de aprender, no es recomendable utilizarla ya que rompe con una de los principios SOLID que deben seguir todas las arquitecturas, específicamente el Single Responsibility, el cual dice que una clase debe tener una sola responsabilidad, en el siguiente fragmento se puede observar como la vista y el controlador están dentro de la misma clase:
-

```
Widget _buildInit() {  
  return Center(  
    child: RaisedButton(  
      child: const Text('Load user data'),  
      onPressed: () {  
        setState(() {  
          _isLoading = true;  
        });  
        widget._repository.getUser().then((user) {  
          setState(() {  
            _user = user;  
            _isLoading = false;  
          });  
        });  
      },  
    ),  
  );  
}
```

Figura 3.19: Uso de la arquitectura de software Vanilla.

Fuente: Platzi

- Scooped Model: Esta arquitectura tiene cierta particularidad y es que cuando hay un cambio de estado en un Widget, este cambio implica una reconstrucción en todo el árbol, por lo tanto no es tan conveniente el uso de esta arquitectura, ya que lo que se requiere es reconstruir un solo Widget sin afectar a los demás.

La ventaja que tiene esta arquitectura es que cumple con el principio de Single Responsibility, ya que separa la lógica del negocio de la UI (Interfaz de usuario), sin embargo, a largo plazo el mantenimiento se vuelve insostenible y complejo por la dependencia que existe entre Widgets.

```
Widget _buildInit(UserModel userModel) {  
  return Center(  
    child: RaisedButton(  
      child: const Text('Load user data'),  
      onPressed: () {  
        userModel.loadUserData();  
      },  
    ),  
  );  
}
```

Figura 3.20: Uso de la arquitectura de software Scooped Model.

Fuente: Platzi

Adicionalmente tenemos la arquitectura BLoC en Flutter, no es más que un patrón de diseño que se encarga de separar la interfaz y el diseño de la lógica del negocio.

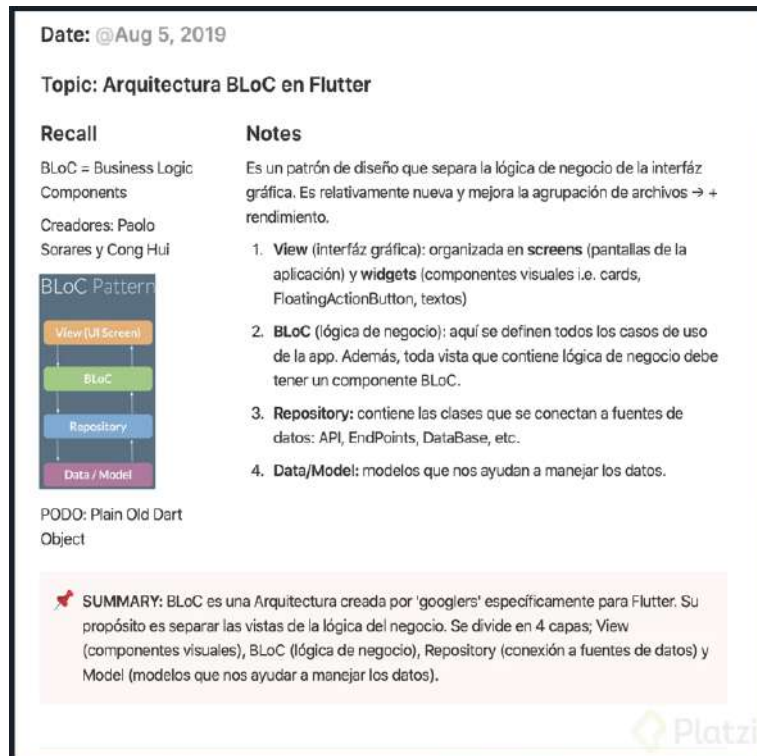


Figura 3.21: Arquitectura BLoC en Flutter.

Fuente: Platzi

Date: @Aug 6, 2019

Topic: BLoC + Clean Architecture en Flutter

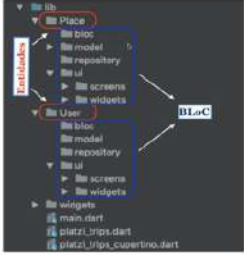
Recall

BLoC + Clean Architecture = código + entendible y mantenible

Notes

- Para que el proyecto sea entendible las entidades son clave.
- Expone las entidades y aplica el patrón de diseño hacia adentro.

Ejemplo:



Aplicando BLoC + Clean Architecture se obtiene:

1. Manejar patrones de diseño para persistencia de datos.
2. Separar la aplicación en diferentes capas.
3. Con un vistazo se puede entender el dominio del proyecto.

SUMMARY: Clean Architecture permite un mejor entendimiento del proyecto mediante la exposición de las entidades del negocio. Además, aplicando BLoC se separa la aplicación en diferentes capas, lo que hace mantenible el código.

Platzi

Figura 3.22: Arquitectura BLoC + Clean Architecture en Flutter.

Fuente: Platzi

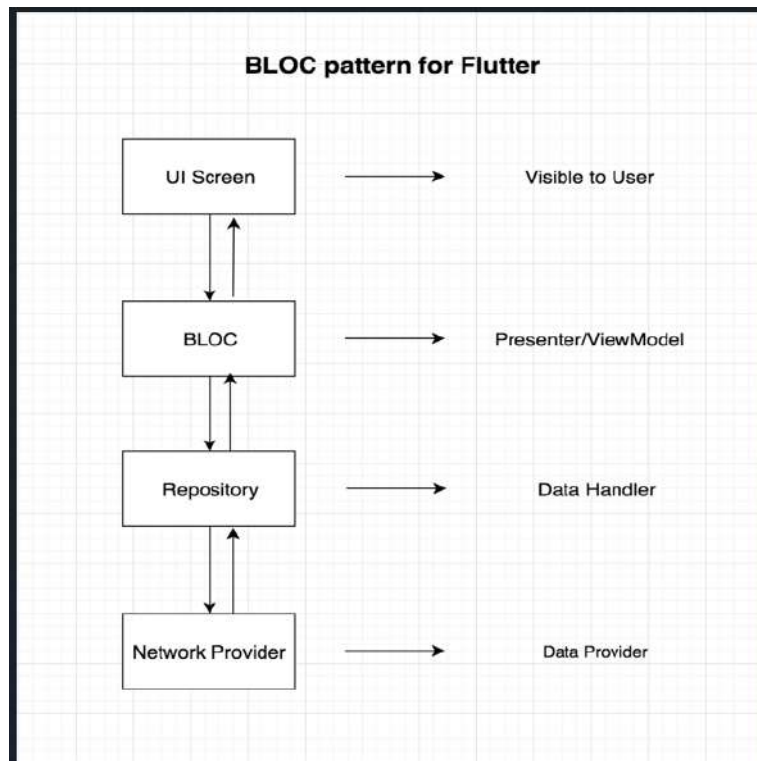


Figura 3.23: Patrón BLoC en Flutter.

Fuente: Platzi

Typescript con Angular

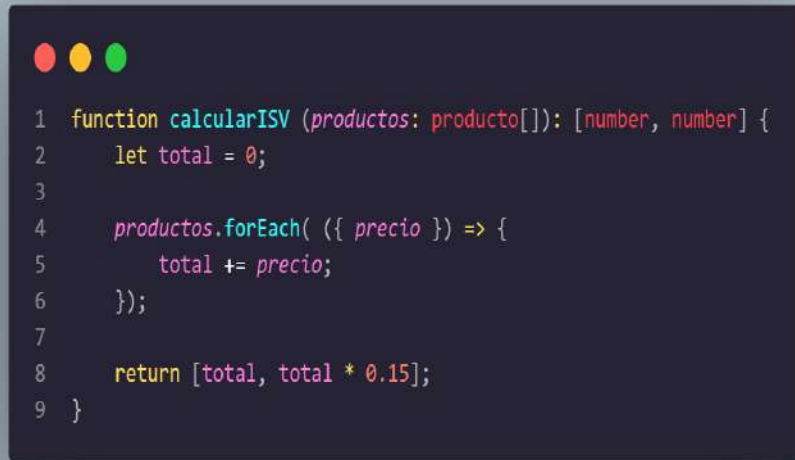
El curso de TypeScript es una introducción a Angular, este framework de JavaScript utiliza TypeScript y más adelante se explicará el por qué. TypeScript se podría considerar un lenguaje de programación para trabajar tanto en el Frontend como en el Backend, a diferencia de muchos lenguajes de programación que generalmente tienen un objetivo específico, como por ejemplo JavaScript para el Frontend, Python para el Backend, entre otros. Este curso le permite al practicante profundizar un poco los conocimientos en JavaScript y adicionarlos en el desarrollo de los proyectos planteados en Angular. Una de las diferencias que existen entre JavaScript y TypeScript es que el primero es un lenguaje de programación no tipado o débilmente tipado, esto quiere decir que podemos asignarle cualquier tipo de valor a nuestras variables, por el contrario, TypeScript es un lenguaje de programación fuertemente tipado, es decir, al momento de declarar una variable, debemos asignarle su tipo.

A screenshot of a code editor with a dark background and light-colored text. The code is a JavaScript function named 'calcularISV' that takes an array 'productos' as an argument. It initializes a 'total' variable to 0, iterates over 'productos' using 'forEach' to calculate the total price, and then returns an array containing the total and a 15% tax amount. The code is numbered from 1 to 9.

```
1 function calcularISV (productos) {  
2   let total = 0;  
3  
4   productos.forEach( ({ precio }) => {  
5     total += precio;  
6   });  
7  
8   return [total, total * 0.15];  
9 }
```

Figura 3.24: Código ejemplo en Javascript.

Fuente: Udemy



```
1 function calcularISV (productos: producto[]): [number, number] {  
2     let total = 0;  
3  
4     productos.forEach( ({ precio }) => {  
5         total += precio;  
6     });  
7  
8     return [total, total * 0.15];  
9 }
```

Figura 3.25: Código ejemplo en Typescript.

Fuente: Udemy

Curso de fundamentos de Angular

Angular es un framework de desarrollo en Javascript, en este punto de la ruta de aprendizaje es importante finalizar con esta tecnología, ya que es la base de la gran mayoría de los proyectos que se requieran próximamente en la empresa, incluso le permite al aprendiz la oportunidad de adquirir conocimientos en otros frameworks de una manera más rápida y eficaz, tales como React, Vue, entre otros.

Este framework es creado y soportado por Google, aproximadamente cada 6 meses se realizan actualizaciones generales del mismo, y es el framework elegido por el equipo de desarrollo de IPT por su robustez y aceptable curva de aprendizaje. El curso permitirá aprender como se da la comunicación de datos con Event Binding, comprender la estructura de un proyecto en angular, conocer las estructuras de NgModel, mejorar la realización de estilos y diseños, crear formularios, etc.

Angular es considerado también una plataforma de desarrollo con múltiples herramientas como PWA, CLI, formularios, etc.



Figura 3.26: Herramientas de Angular.

Fuente: Platzi

Unas de las mejores herramientas que tiene angular al momento de codificar, es la interpolación de variables, funciones o cualquier operador:

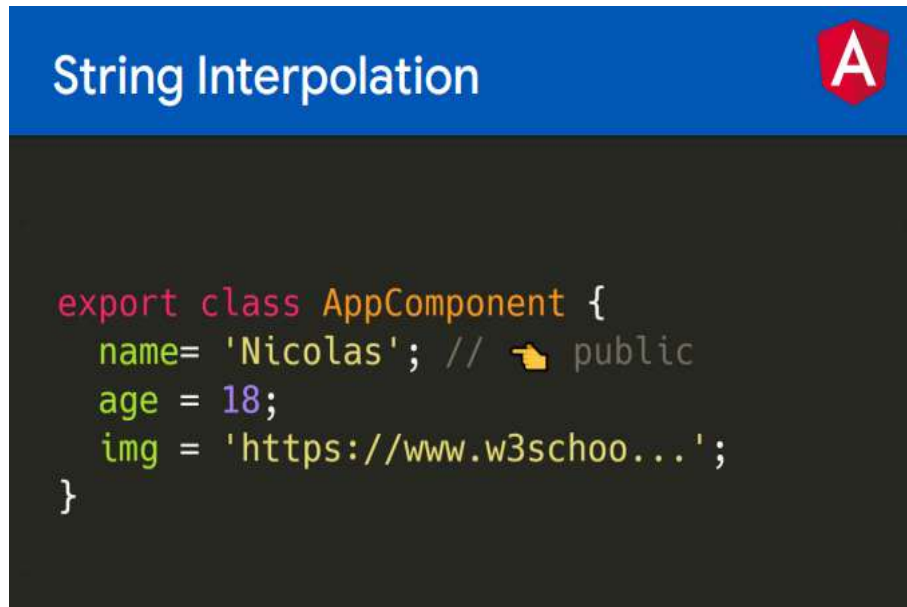


Figura 3.27: Interpolación en Angular.

Fuente: Platzi

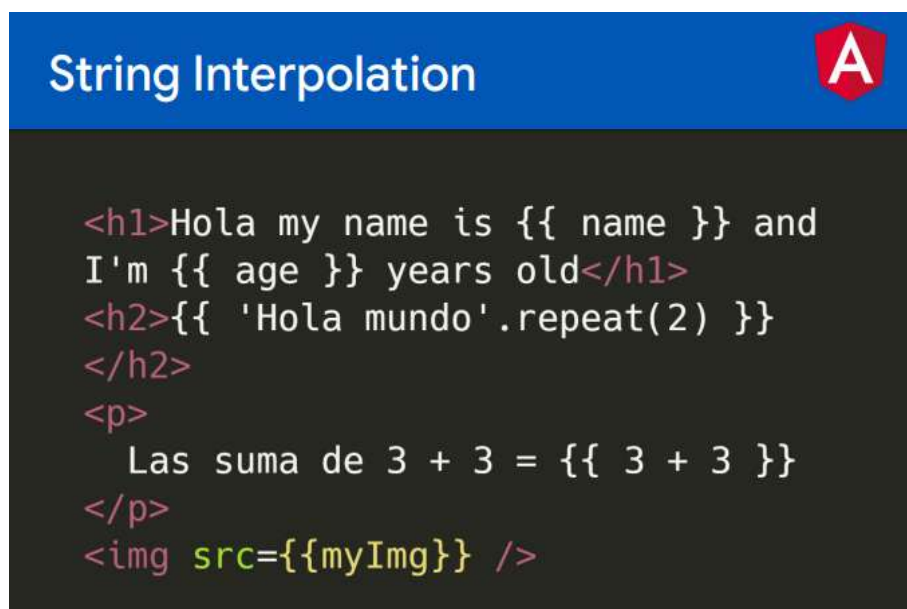


Figura 3.28: Sintaxis de interpolación en Angular.

Fuente: Platzi



Figura 3.29: Sintaxis de NgIf en Angular.

Fuente: Platzi



Figura 3.30: Sintaxis de NgIf - Else en Angular.

Fuente: Platzi



Figura 3.31: Sintáxis de NgFor en Angular.

Fuente: Platzi

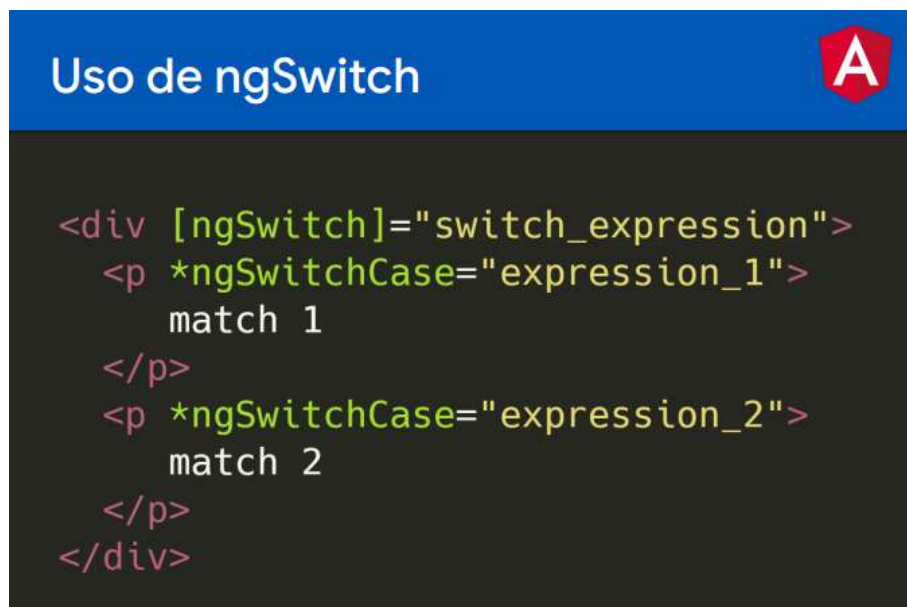


Figura 3.32: Sintáxis de NgSwitch en Angular.

Fuente: Platzi

Curso profesional de Angular

El curso profesional de Angular le permite al aprendiz culminar su ruta de aprendizaje introductoria y profundizar lo aprendido en el curso de fundamentos de Angular, conociendo nuevas terminologías y herramientas que ofrece este maravilloso framework.

Se profundiza en las siguientes herramientas y directivas:

- App Component
 - Métodos en un componente
 - Cambios en el template del componente
 - Directiva NgFor
 - Directiva NgIf
 - Ng-Template
 - FormsModule
 - @Inputs y @Outputs
 - EventEmitter
 - Despliegues a producción
 - Peticiones Http
 - LocalStorage
 - Router Module
 - Formularios reativos
-

3.4 Plataforma para pruebas EKT

En este capítulo se plantea la exposición del objetivo específico relacionado al desarrollo de una plataforma web para pruebas de inglés hechas a la Universidad Francisco de Paula Santander. El desarrollo de esta plataforma web surge por la necesidad de brindar una mejor experiencia a los estudiantes al momento de realizar la prueba, y así mismo ofrecer un servicio mucho más optimizado a la universidad, el practicante se une al equipo de desarrollo de la misma para aportar sus conocimientos en desarrollo frontend y acatar los requisitos propuestos por los PO (Product Owner).

3.4.1 Funcionalidad requerida por la plataforma

Se requiere de una plataforma que permita la creación de usuarios estudiantes de la Universidad Francisco de Paula Santander para que puedan realizar sus exámenes de conocimientos en inglés y puedan cumplir con sus requisitos de grado, la razón de ello es por un convenio pactado por la Academia Inglés Para Todos y la universidad mencionada. A continuación se listan las funcionalidades requeridas por la empresa hacia el equipo de desarrollo:

- Ingreso y registro de estudiantes, se debe tener una vista de Login y Register para que los estudiantes puedan crear usuarios y mantenerlos en base de datos.
 - Videollamadas, es importante que la plataforma cuente con servicio de videollamada para poder monitorear en tiempo real el desarrollo de la prueba por parte de los estudiantes, para ello se debe implementar el uso de una API, más adelante se expondrá mejor esta idea.
 - La plataforma debe contar con filtros que permitan clasificar a los estudiantes que presentan por primera vez la prueba y los que no, esto por reglas de lógica del negocio; para ello se usará lógica de Angular.
 - Debe tener una vista en la que se expliquen los términos y condiciones, reglas de examen y parámetros dictados por el supervisor de la prueba, al final de esta vista debe haber una sección en la que el estudiante pueda agendar el día y hora de su prueba, ofreciendo así un servicio optimizado y sencillo para los estudiantes.
 - La plataforma debe contar con anuncios de todas las convocatorias vigentes y por ofertarse.
 - Entre unas de las secciones, debe haber un videotutorial que explique el paso a paso sobre como realizar correctamente la prueba cumpliendo con los patrones de desarrollo y lógica de negocio de la empresa.
-

- Es importante contar con un sistema anti-fraude para monitorear a los estudiantes y evitar la ejecución de trampas por parte de ellos, tales como uso de pantallas externas o programas de terceros para obtener ayuda, para ello se utilizarán herramientas ofrecidas por Java.
- Así mismo la plataforma debe contar con el examen online, para ello se debe realizar un microservicio que contemple esa prueba y su respectivo banco de preguntas.
- Es fundamental que la plataforma cuente con una sección para el administrador, la cual permita el acceso al historial de pruebas y resultados de todos los estudiantes.
- La plataforma debe contar con una base de datos para generar los reportes de pruebas y pasarlos al equipo de marketing de la empresa.

3.4.2 Elaboración de historias de usuarios

Antes de describir las historias de usuario elaboradas, vamos a definir primeramente que es una historia de usuario. Las historias de usuario se pueden entender como cortas descripciones que tienden a ser simples y con una característica en específico la cual se cuenta y expone desde la perspectiva del usuario que requiere una determinada funcionalidad que generalmente es un cliente del sistema o simplemente un usuario. Estas historias de usuario cuentan con tres parámetros, un "Como" que en este caso sería una condición del "Usuario", un "Quiero" el cual sería el objetivo del "Usuario" y un "Para" que vendría siendo el motivo por el cual se requiere esa funcionalidad. [15]

Teniendo claro el concepto de historias de usuario, a continuación se exponen las historias de usuario realizadas por el equipo de desarrollo:

HU 0001	
COMO	Usuario de ADMIN
QUIERO	Poder visualizar todas los grupos disponibles dentro de una convocatoria de pruebas EKT.
PARA	Poder unirme con todos los privilegios de anfitrión a la aplicación de videollamada que se estuvo implementando. Poder diferenciar en la videollamada usuarios de maestría y usuarios de pregrado. Poder visualizar el listado y la cantidad de participantes inscritos en un grupo.
PRIORIDAD	Alta
PUNTAJE	8

Figura 3.33: Historia de usuario 0001.

Fuente: Autor

HU 0002	
COMO	Usuario de ADMIN
QUIERO	Poder visualizar a los estudiantes durante la presentación de una prueba.
PARA	Poder visualizar en tiempo real su desempeño en la prueba, flags y alertas que el algoritmo antifraude arroja. Poder anular a los estudiantes que detecte haciendo fraude o irregularidades.
PRIORIDAD	Alta
PUNTAJE	5

Figura 3.34: Historia de usuario 0002.

Fuente: Autor

HU 0003	
COMO	Usuario de ADMIN
QUIERO	Tener un buscador del historial estudiantil.
PARA	Poder visualizar los grupos a los cuales fue reprogramado el estudiante. Poder visualizar los resultados de las pruebas que el estudiante presentó en el sistema con día, fecha y hora.
PRIORIDAD	Alta
PUNTAJE	8

Figura 3.35: Historia de usuario 0003.

Fuente: Autor

HU 0004	
COMO	Usuario estudiante
QUIERO	Poder vincularme a una convocatoria.
PARA	Poder visualizar mi próxima prueba agendada de forma rápida y clara. Poder visualizar el estado de convocatorias ordenadas por fechas. Poder reconocer que me he registrado correctamente en una convocatoria. Poder aceptar los términos y condiciones e inscribirse a una convocatoria en la que esté interesado.
PRIORIDAD	Muy alta
PUNTAJE	8

Figura 3.36: Historia de usuario 0004.

Fuente: Autor

HU 0005	
COMO	Usuario estudiante
QUIERO	Tener un perfil
PARA	Poder editar mi información personal. Poder ver mi historial académico con nota.
PRIORIDAD	Media
PUNTAJE	3

Figura 3.37: Historia de usuario 0005.

Fuente: Autor

HU 0006	
COMO	Usuario externo universidad
QUIERO	Tener un perfil de consultas
PARA	Poder visualizar los grupos a los cuales fue reprogramado el estudiante. Poder visualizar los resultados de las pruebas que el estudiante presentó en el sistema con día, fecha y hora.
PRIORIDAD	Media
PUNTAJE	2

Figura 3.38: Historia de usuario 0006.

Fuente: Autor

3.4.3 Ejecución del proyecto

Dentro del equipo de desarrollo se planteó una metodología de pequeñas tareas a realizar respecto a cada historia de usuario las cuales llamamos "cpp's". Esto se hace con el fin de dividir tareas y tener un buen desarrollo de las HU para el proyecto.

Como toda plataforma necesitamos tener una vista de login, olvido de contraseña y registro para usuarios nuevos.

Login

Este código representa toda la lógica que tiene internamente el login, contiene las importaciones de librerías necesarias, entradas y salidas para la comunicación de servicios y un formulario que contiene básicamente tres validaciones, es decir, se recibe número de documento, código de la universidad y contraseña.

Código 3.1: Código .ts (TypeScript) del login - plataforma EKT

```
1 import { Component, Output, EventEmitter, Input } from '@angular/core';
2 import { FormBuilder, FormControl, FormGroup, Validators } from '@angular/forms';
3 import { Session } from 'src/app/core/models/session/session.model';
4 import { LoginRq } from 'src/app/core/models/user/user-rq.model';
5 import { UserRs } from 'src/app/core/models/user/user-rs.model';
6 import { StorageService } from 'src/app/core/services/storage/storage-
7   service';
8 import { UserService } from 'src/app/core/services/user/user-service';
9
10 @Component({
11   selector: 'app-login',
12   templateUrl: './login.component.html',
13   styleUrls: ['./login.component.css']
14 })
15 export class LoginComponent {
16
17   @Input () copies: any;
18   @Output() login = new EventEmitter<boolean>();
19   @Output() formType = new EventEmitter<number>();
20
21   loginForm!: FormGroup;
22   loader: boolean = false;
23   user!: UserRs;
24
25   constructor(
26     fb: FormBuilder,
```

```
27 private storageService: StorageService ) {
28   this.loginForm = fb.group({
29     ufpsCode: ['', [Validators.required, Validators.minLength(7), ↵
        ↵ Validators.maxLength(7)]]],
30     idCard: ['', [Validators.required, Validators.minLength(5), ↵
        ↵ Validators.maxLength(20)]]],
31     password: ['', [Validators.required, Validators.minLength(6), ↵
        ↵ Validators.maxLength(32)]]],
32   });
33 }
34
35 actionButton() {
36   this.loader = true;
37
38   let request: LoginRq = {
39     id_card: this.loginForm.controls.idCard.value,
40     ufps_code: this.loginForm.controls.ufpsCode.value,
41     password: this.loginForm.controls.password.value,
42   }
43
44   this.userService.login(request).subscribe( response => {
45     switch (response.message.code) {
46       case 1:
47         const login: Session = {
48           token: response.data.token,
49           user: response.data,
50         }
51         this.storageService.setCurrentSession(login);
52         this.login.emit(true);
53         break;
54       case 2:
55         alert('Los datos ingresados no corresponden a un usuario ↵
        ↵ registrado en nuestro sistema, por favor reviselos.');
```

```

67
68   clickedLink(event: number){
69       this.formType.emit(event);
70   }
71
72 }

```

Olvido de contraseña

La interfaz para restablecer la contraseña contiene un campo menos que el login, claro está porque lo que se busca es restablecer la respectiva contraseña.

Código 3.2: Código .ts (TypeScript) de la sección "Olvido de contraseña" - plataforma EKT

```

1 import { Component, EventEmitter, Input, Output } from '@angular/core';
2 import { FormBuilder, FormGroup, Validators } from '@angular/forms';
3 import { ResetPasswordRq } from 'src/app/core/models/user/user-rq.model' ←
   → ;
4 import { UserService } from 'src/app/core/services/user/user-service';
5
6 @Component({
7   selector: 'app-forgot-password',
8   templateUrl: './forgot-password.component.html',
9   styleUrls: ['./forgot-password.component.css']
10 })
11 export class ForgotPasswordComponent {
12
13   @Input () copies: any;
14   @Output() formType = new EventEmitter<number>();
15
16   forgotForm!: FormGroup;
17   loader: boolean = false;
18
19   constructor(
20     fb: FormBuilder,
21     private userService: UserService ) {
22     this.forgotForm = fb.group({
23       ufpsCode: ['', [Validators.required, Validators.minLength(7), ←
   → Validators.maxLength(7)]],
24       idCard: ['', [Validators.required, Validators.minLength(5), ←
   → Validators.maxLength(20)]]
25     });
26   }
27
28   actionButton() {

```

```
29   this.loader = true;
30
31   let request: ResetPasswordRq = {
32     id_card: this.forgotForm.controls.idCard.value,
33     ufps_code: this.forgotForm.controls.ufpsCode.value,
34   }
35
36   this.userService.resetPassword(request).subscribe( data => {
37     this.showAlert(data)
38     this.loader = false;
39   });
40 }
41
42
43 showAlert(apiResponse: any) {
44   switch (apiResponse.message.code) {
45     case 1:
46       alert(apiResponse.data);
47       this.backLogin();
48       break;
49     case 3:
50       alert('Los parametros ingresados corresponden a un usuario que ↩
51         ↩ no existe.');
```

```
51       break;
52     case 2:
53       alert(apiResponse.message);
54       break;
55     default:
56       alert('Ha ocurrido un error inesperado.');
```

```
57     break;
58   }
59 }
60
61 backLogin(){
62   this.formType.emit(0);
63 }
64
65 }
```

Registro para usuarios nuevos

La sección de registro contiene diversos campos que son validados como se muestra en el siguiente código, conteniendo los respectivos llamados a los servicios principales de la API.

Código 3.3: Código .ts (TypeScript) de la sección "Creación de cuenta nueva" - plataforma EKT

```

1 import { Component, EventEmitter, Input, Output, SimpleChanges } from '@angular/core';
2 import { AbstractControl, EmailValidator, FormBuilder, FormGroup, ValidationErrors, Validators } from '@angular/forms';
3 import { UserRq } from 'src/app/core/models/user/user-rq.model';
4 import { UserService } from 'src/app/core/services/user/user-service';
5
6 @Component({
7   selector: 'app-register',
8   templateUrl: './register.component.html',
9   styleUrls: ['./register.component.css']
10 })
11 export class RegisterComponent {
12
13   @Input() copies: any;
14   @Output() formType = new EventEmitter<number>();
15
16   registerForm!: FormGroup;
17   labels: any;
18   loader: boolean = false;
19
20   constructor(
21     fb: FormBuilder,
22     public userService: UserService) {
23     this.registerForm = fb.group({
24       name: ['', [Validators.required, Validators.minLength(3), Validators.maxLength(300)]],
25       lastName: ['', [Validators.required, Validators.minLength(3), Validators.maxLength(300)]],
26       phone: ['', [Validators.required, Validators.minLength(10), Validators.maxLength(15)]],
27       ufpsCode: ['', [Validators.required, Validators.minLength(7), Validators.maxLength(7)]],
28       idCard: ['', [Validators.required, Validators.minLength(5), Validators.maxLength(20)]],
29       password: ['', [Validators.required, Validators.minLength(6), Validators.maxLength(32)]],
30       email: ['', [Validators.required, Validators.maxLength(300), this.userService]]

```

```
        ↪ dominio]],
31    rePassword: ['', [Validators.required, Validators.minLength(6), ↪
        ↪ Validators.maxLength(32)]],
32    });
33  }
34
35  this.userService.register(request).subscribe( data => {
36    this.showAlert(data.message.code)
37    this.loader = false;
38  });
39  }
40 }
```

Teniendo desarrollados las tres secciones principales, pasamos a todo lo que tiene que ver con funcionalidades internas, dashboard, vista de home tanto para instructores como estudiantes, tenemos una sección de convocatoria de pruebas, edición de perfil para los estudiantes, seguimiento y monitoreo de pruebas en tiempo real para el supervisor, vista para la información de pruebas por grupos y horario y funcionalidades generales como ingreso a la videollamada y prueba.

Home

La vista del dashboard principal es maquetada en gran parte por el aprendiz, su contribución fue considerable y notable en el desarrollo de la vista principal, realizando un carrusel de imágenes que hacían representación de todas las convocatorias, un carrusel incorporado netamente con HTML y CSS, claramente se pudo haber realizado con un poco de Javascript, sin embargo la exigencia por tiempos para presentar el producto era alta, adicionalmente el tiempo era corto, la vista de home contiene a su vez una lógica interesante para mostrar un video tutorial y mostrar los pasos para presentar una prueba EKT.

Código 3.4: Código .ts (TypeScript) del home - plataforma EKT

```
1 import { Component, EventEmitter, Input, Output } from '@angular/core';
2
3 @Component({
4   selector: 'app-home',
5   templateUrl: './home.component.html',
6   styleUrls: ['./home.component.css']
7 })
8 export class HomeComponent {
9
10  @Input() userToken!: string;
11  @Input() blockGroupTime!: boolean;
12  @Output() pageChange = new EventEmitter<string>();
13
14
15  constructor() { }
16
17  carouselAction() {
18    this.pageChange.emit('announcement');
19  }
20
21  pageActionChange() {
22    this.pageChange.emit('meeting');
23  }
24
25 }
```

Grupos EKT

La vista de grupos es una vista netamente para el usuario docente y el usuario supervisor, es de gran importancia ya que permite visualizar el listado de los estudiantes divididos por secciones horarias, es una vista importante ya que se puede hacer un seguimiento previo de los estudiantes que van a presentar determinada prueba, revisar sus datos y la cantidad de los participantes, así mismo tiene al final un botón que permite dar ingreso a esa video llamada. Esta vista tiene implementada una tabla desarrollada por Bootstrap, el aprendiz se encargó de traer los datos mediante el servicio de la API, estructurando bien las columnas de la tabla y aplicándole los estilos que se requerían para poder combinar con los colores de la página.

Código 3.5: Código .ts (TypeScript) de la sección de "Grupos EKT" - plataforma EKT

```
1 import { Component, OnInit, EventEmitter, Output } from '@angular/core'↵
2   ↵';
3 import { GroupService } from 'src/app/core/services/group/group-service'↵
4   ↵';
5
6
7 @Component({
8   selector: 'app-groups-ekt',
9   templateUrl: './groups-ekt.component.html',
10  styleUrls: ['./groups-ekt.component.css']
11 })
12 export class GroupsEktComponent {
13
14   @Output() pageChange = new EventEmitter<string>();
15   data_groups: any[] = [];
16   room!: string;
17
18   constructor(private groupService: GroupService) { }
19
20   executeQuery(event: string) {
21     this.groupService.getUserPerGroup(event).subscribe( (response) => {
22       if (response.message.code == 1) {
23         this.data_groups = response.data;
24       }
25     });
26   };
27
28   accionElemento(data: any) {
29     console.log('principal', data);
30   }
31
32   enterIntoMeet(){
```

```

31  var form = document.createElement('form');
32  document.body.appendChild(form);
33  form.method = 'get';
34  form.target = '_blank';
35  form.action = "meeting";
36
37  form.submit();
38  }
39
40  }

```

Puntajes EKT

La vista de puntajes EKT es muy parecida a la de grupos EKT, tiene el mismo buscador para seleccionar el día y hora de la prueba, fue apoyada técnicamente por el aprendiz también, y su diferencia con la vista de grupos es que en esta se puede visualizar las preguntas correctas e incorrectas que llevan los estudiantes, adicionalmente tiene un botón que permite anular la prueba en caso de que se presenten fraudes o irregularidades que se anuncian previamente.

Código 3.6: Código .ts (TypeScript) de la sección de "Score EKT" - plataforma EKT

```

1 import { Component, EventEmitter, Output , Input, OnInit } from '@angular/core';
2 import { GroupService } from 'src/app/core/services/group/group-service';
3
4 @Component({
5   selector: 'app-ekt',
6   templateUrl: './ekt.component.html',
7   styleUrls: ['./ekt.component.css']
8 })
9 export class EktComponent {
10
11   query: string = '';
12   data_score: any = [];
13
14   @Input () copies: any;
15   @Output() pageChange = new EventEmitter<string>();
16
17   constructor(private groupService: GroupService) { }
18
19   carouselAction() {
20     this.pageChange.emit('announcement');
21   }

```

```
22
23 processQuery(event: string){
24   this.query = event;
25   setInterval( () => {
26     console.log('updatee');
27     this.groupService.getRealtimeData(this.query).subscribe( (response) => {
28       if (response.message.code == 1) {
29         this.data_score = response.data;
30       }
31     });
32   }, 7500);
33   this.getRealtimeData(event);
34 }
35
36 // Query debe ser un string como el ejemplo '2021-10-25/08:30:00'
37 getRealtimeData(query: string) {
38   this.groupService.getRealtimeData(query).subscribe( (response) => {
39     if (response.message.code == 1) {
40       this.data_score = response.data;
41     }
42   });
43 }
44
45
46 }
```

3.4.4 Finalización y retroalimentación

El desarrollo de todas las historias de usuario se pudo completar en tiempo record, obteniendo buenos resultados sobre todo en la incorporación de la API de Jitsi para las videollamadas, la plataforma EKT cumplió con las expectativas y tuvo un buen rendimiento durante la ejecución de todas las convocatorias de pruebas por parte de los estudiantes de la UFPS, todo el código fue desarrollado en equipo y con tareas divididas, así mismo se hizo un buen manejo de git y github para la creación del microservicio como repositorio en la plataforma de github y paso a producción por medio del servidor usado en ese momento.

Como retroalimentación, podemos hacer énfasis en falencias al ser el primero proyecto realizado y desarrollado por un equipo relativamente nuevo de desarrollo, estos puntos negativos concluidos por el equipo son claves para el mejoramiento de desarrollo en los próximos proyectos requeridos por la empresa, con ello se fomenta el buen aprendizaje a base de errores.

3.5 IPT World - App móvil

Este proyecto es orientado principalmente para los estudiantes de la Academia Inglés Para Todos, se busca desarrollar una aplicación móvil multiplataforma con didáctica de juegos para niños y jóvenes. Se optó por elegir Flutter como framework de desarrollo por su corta curva de aprendizaje y compatibilidad multiplataforma, lo cual era uno de los objetivos principales por el product owner.

La estructura del proyecto se compone profundamente de tres particiones que forman el corazón de la aplicación: componentes, núcleos y pages o páginas.

3.5.1 Componentes

Aquí se continúa un proceso que es tendencia en el equipo, planificar los proyectos con metodologías que hagan parte de antiguos proyectos, en este caso se sigue optando de la metodología atomic design incluso para el desarrollo de Apps, esto se hace con el fin de llevar el mismo orden y profesionalismo a la hora de elaborar proyectos. El concepto es simple, partimos desde componentes más pequeños hasta los más grandes, es útil usar la metodología atomic design porque nos permite realizar una buena analogía del tamaño de componentes de un proyecto, en este caso iniciamos desde átomos, para componentes mayores usamos moléculas ya que son un conjunto de átomos, para componentes muchos más grandes que moléculas usamos organismos, los cuales son composiciones de moléculas; y finalmente, determinamos los componentes más grandes como plantillas o páginas, algunas veces se utilizan plantillas y las páginas las denominamos grupos de plantillas.

Atomos

- FormFields.dart
 - Buttons.dart
 - Colors.dart
 - Containers.dart
 - Images.dart
 - Inputs.dart
 - Labels.dart
 - Shapes.dart
 - Spaces.dart
 - Text.dart
-

Moléculas

- Backgrounds.dart
- Forms.dart
- Inputs.dart
- List.dart
- Menu.dart
- Padding.dart
- Positioned.dart
- Section.dart

Organismos

- Covers.dart
- Navigation-bar.dart

Plantillas

- index-template.dart
 - level-info-template.dart
 - liberty-template.dart
 - login-template.dart
 - pre-level-template.dart
 - register-template
-

3.5.2 Núcleos

Los core se encargan de contener preprocesadores de la aplicación como modelos y servicios que nos permiten conectar peticiones y enlazar comunicaciones mediante la web.

Blocs

- User-bloc.dart

Código 3.7: Código .dart del core user-bloc - IPT World

```
1 import 'package:rxdart/rxdart.dart';
2 import 'package:ipt_world/src/core/services/user-service.dart';
3 import 'package:ipt_world/src/utils/validators.dart';
4
5 class UserBloc with Validators{
6
7   final userService = new UserService();
8
9   final _emailController = BehaviorSubject<String>();
10
11   Stream<String> get emailStream => _emailController.stream.transform(↵
    ↵ validateEmail);
12   Stream<String> get phoneStream => _emailController.stream.transform(↵
    ↵ validateEmail);
13   Stream<String> get nameStream => _emailController.stream.transform(↵
    ↵ validateEmail);
14   Stream<String> get lastNameStream => _emailController.stream.↵
    ↵ transform(validateEmail);
15
16   Function(String) get changeEmail => _emailController.sink.add;
17   String get email => _emailController.value;
18
19   dispose() {
20     _emailController?.close();
21   }
22
23 }
```

Interfaces

- media.dart

Código 3.8: Código .dart de la interfaz "media" - IPT World

```
1 class MediaInterface {
2
3   //Object properties same at fuction inputs
4   double _height;
5   double _width;
6
7   MediaInterface(
8     this._height,
9     this._width
10  );
11
12  get height => this._height;
13  set height(double height){
14    _height = height;
15  }
16
17  get width => this._width;
18  set width(double width){
19    _width = width;
20  }
21
22 }
```

Modelos

- Levels.dart

Código 3.9: Código .dart del modelo de niveles - IPT World

```
1 class Levels{
2
3   //Object properties same at database
4   int _code;
5   String _name;
6   String _color;
7   String _assetCard;
8   String _assetCharacter;
9
10  //Constructor with all properties
11  Levels(
12    this._code,
13    this._name,
14    this._color,
15    this._assetCard,
16    this._assetCharacter
17  );
18
19  //Getters y setters
20  get code => this._code;
21  set code(int code){
22    _code = code;
23  }
24
25  get name => this._name;
26  set name(String name){
27    _name = name;
28  }
29
30  get color => this._color;
31  set color(String color){
32    _color = color;
33  }
34
35  get assetCard => this._assetCard;
36  set assetCard(String assetCard){
37    _assetCard = _assetCard;
38  }
39
40  get assetCharacter => this._assetCharacter;
```

```
41 set assetCharacter(String assetCharacter){
42     _assetCharacter = assetCharacter;
43 }
44
45 //Mapping to json
46 Map<String, dynamic> toJson() => {
47     'code': _code,
48     'name': _name,
49     'asset_card': _assetCard,
50     'asset_character': _assetCharacter
51 };
52
53 //Serialize from json
54 Levels.fromJson(Map<String, dynamic> json) {
55     _code = json["code"];
56     _name = json["name"];
57     _assetCard = json["assetCard"];
58     _assetCharacter = json["assetCharacter"];
59 }
60
61 static List<Levels> getMockLevels() {
62     List<Levels> levels = [];
63
64     levels.add(new Levels(0, "Vlad's Dimension", "vampire-color" , "↵
        ↵ assets/images/cards/vampire.png", "assets/images/characters/↵
        ↵ vampire.png"));
65     levels.add(new Levels(1, "Anastasia's Crystal", "anastasya-color", "↵
        ↵ assets/images/cards/lady.png", "assets/images/characters/lady↵
        ↵ .png"));
66     levels.add(new Levels(2, "Vicco's Lab", "vicco-color", "assets/↵
        ↵ images/cards/scientific.png", "assets/images/characters/↵
        ↵ scientific.png"));
67     levels.add(new Levels(3, "Hikari's Mountain", "hikari-color", "↵
        ↵ assets/images/cards/lynx.png", "assets/images/characters/lynx↵
        ↵ .png"));
68     levels.add(new Levels(4, "Delta's City", "delta-color", "assets/↵
        ↵ images/cards/robot.png", "assets/images/characters/robot.png↵
        ↵ "));
69     levels.add(new Levels(5, "Keplerian Planet", "keplerian-color", "↵
        ↵ assets/images/cards/alien.png", "assets/images/characters/↵
        ↵ alien.png"));
70
71     return levels;
72 }
73
```

74 }

- Station.dart

Código 3.10: Código .dart del modelo de estación - IPT World

```
1 class Station {
2
3   //Object properties same at database
4   int _code;
5   String _name;
6   int _stage;
7   int _appLevel;
8
9   //Constructor with all properties
10  Station(
11    this._code,
12    this._name,
13    this._stage,
14    this._appLevel
15  );
16
17  //Getters y setters
18  get code => this._code;
19  set code(int code){
20    _code = code;
21  }
22
23  get name => this._name;
24  set name(String name){
25    _name = name;
26  }
27
28  get stage => this._stage;
29  set stage(int stage){
30    _stage = _stage;
31  }
32
33  get appLevel => this._appLevel;
34  set appLevel(int appLevel){
35    _appLevel = appLevel;
36  }
37
38  //Mapping to json
39  Map<String, dynamic> toJson() => {
40    'code': _code,
41    'name': _name,
```

```
42     'stage': _stage,
43     'app_level': _appLevel
44 };
45
46 //Serialize from json
47 Station.fromJson(Map<String, dynamic> json) {
48     _code = json['code'];
49     _name = json['name'];
50     _stage = json['stage'];
51     _appLevel = json['app_level'];
52 }
53
54 }
```

3.5.3 Páginas

Actividades

- t-io-page.dart

Código 3.11: Código .dart de actividad 1 - IPT World

```
1 import 'package:flutter/material.dart';
2
3 class TioPage extends StatelessWidget {
4
5   @override
6   Widget build(BuildContext context) {
7     return Container(
8       child: null,
9     );
10  }
11
12 }
```

- t-ip-page.dart

Código 3.12: Código .dart de actividad 2 - IPT World

```
1 import 'dart:async';
2
3 import 'package:flutter/material.dart';
4 import 'package:flutter/services.dart';
5 import 'package:ipt_world/src/components/atoms/colors.dart';
6 import 'package:ipt_world/src/components/atoms/images.dart';
7 import 'package:ipt_world/src/components/atoms/labels.dart';
8 import 'package:ipt_world/src/components/templates/activities/t-↵
  ↵ ip_template.dart';
9 import 'package:ipt_world/src/core/interfaces/media.dart';
10 import 'package:ipt_world/src/core/models/Station.dart';
11
12 class TipPage extends StatefulWidget {
13
14   @override
15   _TipPageState createState() => _TipPageState();
16 }
17
18 class _TipPageState extends State<TipPage> {
19
```

```

20  MediaInterface _screen;
21  List<Color> _colors;
22  List<List<Widget>> _cardOptions;
23  List<int> _cardState = [0,0,0,0,0,0,0,0];
24  List<String> _objects = ['Tiger', 'assets/images/tipshark.png', 'Cat'↵
    ↵ ', 'assets/images/tiplion.png', 'Lion', 'assets/images/tipcat.↵
    ↵ png', 'Shark', 'assets/images/tiptiger.png'];
25
26
27  @override
28  dispose() {
29    SystemChrome.setPreferredOrientations([DeviceOrientation.portraitUp↵
    ↵ ]);
30    super.dispose();
31  }
32
33  @override
34  Widget build(BuildContext context) {
35    SystemChrome.setPreferredOrientations([
36      DeviceOrientation.landscapeRight,
37      DeviceOrientation.landscapeLeft
38    ]);
39
40    _screen = MediaInterface(MediaQuery.of(context).size.height, ↵
    ↵ MediaQuery.of(context).size.width);
41
42    Station station = ModalRoute.of(context).settings.arguments;
43
44    // TO - DO usar el station pa obtener preguntas del server
45
46    _colors = ColorEnum.getThemeArrayColors(station.appLevel);
47    _cardOptions = _setChildrens();
48    List<Widget> childrens = _getChildrens();
49
50    return Scaffold(
51      backgroundColor: ColorEnum.white,
52      body: TipTemplate.page(context, _screen, station.appLevel, _colors↵
    ↵ , childrens)
53    );
54  }
55
56  List<Widget> _getChildrens() {
57    List<Widget> response = [];
58
59    for(int i=0; i<_cardOptions.length; i++) {

```

```

60     if(_cardState[i]==0 || _cardState[i]==1) {
61         response.add(_actionCard(i, _cardOptions[i][_cardState[i]]));
62     } else {
63         response.add(Labels.tipChecked(_screen.width*0.1, _colors[2]));
64     }
65 }
66
67 return response;
68 }
69
70 List<List<Widget>> _setChildrens() {
71     List<List<Widget>> response = [];
72     String path = 'assets/images/tip';
73
74     for(int i=0; i<_objects.length; i++){
75         List<Widget> options = [];
76         options.add(Labels.colorButtonLight(_screen.width*2.5, '?', ↩
77             ↩ _colors[2]));
78         if(_objects[i].contains(path)){
79             options.add(Images.mapImage(_objects[i]));
80         } else {
81             options.add(Labels.colorButtonLight(_screen.width, _objects[i], ↩
82                 ↩ _colors[2]));
83         }
84         response.add(options);
85     }
86     return response;
87 }
88
89 Widget _actionCard(int i, Widget children) {
90     return InkWell(
91         child: children,
92         onTap: () {
93             // ignore: unrelated_type_equality_checks
94             if(_cardState[i] == 0 && !_existTwoCardBack()) {
95                 _cardState[i] = 1;
96                 setState(() {});
97             }
98
99             if(_existTwoCardBack()){
100                 print('enter hear');
101
102                 List<int> selected = [];
103                 for(int i=0; i<_cardState.length; i++) {

```

```
103         if(_cardState[i] == 1){
104             selected.add(i);
105         }
106     }
107
108     _validateCorrectOptions(selected);
109
110     print(selected);
111 }
112
113     print('ontap action $i');
114     print(_cardState);
115 },
116 );
117 }
118
119 bool _existTwoCardBack(){
120     int count = 0;
121     for(int i=0; i<_cardState.length; i++){
122         count += _cardState[i] == 1 ? 1 : 0;
123     }
124     return count == 2;
125 }
126
127 void _validateCorrectOptions(List<int> selected) {
128     String a = _objects[selected[0]].toLowerCase();
129     String b = _objects[selected[1]].toLowerCase();
130
131
132     int value = a.contains(b) || b.contains(a) ? 3 : 0;
133     Timer(new Duration(milliseconds: 750), () {
134         _cardState[selected[0]] = value;
135         _cardState[selected[1]] = value;
136         setState(() {});
137     });
138 }
139
140
141 }
```

- t-oo-page.dart

Código 3.13: Código .dart de actividad 3 - IPT World

```
1 import 'package:flutter/material.dart';
2
3 class TooPage extends StatelessWidget {
4
5   @override
6   Widget build(BuildContext context) {
7     return Container(
8       child: null,
9     );
10  }
11
12 }
```

Vista de autenticación

Código 3.14: Código .dart de vista de autenticación - IPT World

```

1 import 'package:flutter/material.dart';
2
3 import 'package:ipt_world/src/components/atoms/colors.dart';
4 import 'package:ipt_world/src/components/organisms/navigation_bar.dart ↵
   ↵ ';
5 import 'package:ipt_world/src/components/templates/index_template.dart ↵
   ↵ ';
6 import 'package:ipt_world/src/core/models/Levels.dart';
7
8
9 class AuthPage extends StatefulWidget {
10
11   @override
12   _AuthPageState createState() => _AuthPageState();
13
14 }
15
16 class _AuthPageState extends State<AuthPage> {
17
18   int _levelInt = 0;
19
20   @override
21   Widget build(BuildContext context) {
22
23     final double _width = MediaQuery.of(context).size.width;
24     final double _height = MediaQuery.of(context).size.height;
25     List<Levels> levels = Levels.getMockLevels();
26
27     return Scaffold(
28       backgroundColor: ColorEnum.white,
29       body: IndexTemplate.content(context, _height, _width, "Juan Camilo ↵
   ↵ $_levelInt", "Beginner", "assets/images/characters/vaquera ↵
   ↵ .png", levels),
30       bottomNavigationBar: NavigationBar.navigationBar(context, _height, ↵
   ↵ _width, 1, ColorEnum.vampire(0.6)),
31     );
32   }
33
34 }

```

Vista de inicio

Código 3.15: Código .dart de vista de inicio - IPT World

```
1 import 'package:flutter/material.dart';
2 import 'package:ipt_world/src/components/templates/liberty_template.dart';
3 import 'package:ipt_world/src/components/templates/login_template.dart';
4
5 class HomePage extends StatelessWidget {
6
7   @override
8   Widget build(BuildContext context) {
9     final double _width = MediaQuery.of(context).size.width;
10    final double _height = MediaQuery.of(context).size.height;
11
12    return Scaffold(
13      backgroundColor: Color.fromRGB(255, 255, 255, 1),
14      body: PageView(
15        scrollDirection: Axis.vertical,
16        children: <Widget>[
17          LibertyTemplate.stackPage(_height, _width),
18          LoginTemplate.scrollPage(_height, _width, context),
19        ],
20      ),
21    );
22  }
23
24 }
```

Vista de mapa

Código 3.16: Código .dart de vista de mapas - IPT World

```

1 import 'package:flutter/material.dart';
2
3 import 'package:ipt_world/src/components/templates/maps/↵
  ↵ anastasya_template.dart';
4 import 'package:ipt_world/src/components/templates/maps/delta_template.↵
  ↵ dart';
5 import 'package:ipt_world/src/components/templates/maps/hikari_template↵
  ↵ .dart';
6 import 'package:ipt_world/src/components/templates/maps/↵
  ↵ keplerian_template.dart';
7 import 'package:ipt_world/src/components/templates/maps/vicco_template.↵
  ↵ dart';
8 import 'package:ipt_world/src/components/templates/maps/vlad_template.↵
  ↵ dart';
9
10 class MapPage extends StatelessWidget {
11   @override
12   Widget build(BuildContext context) {
13
14     final double _width = MediaQuery.of(context).size.width;
15     final double _height = MediaQuery.of(context).size.height;
16     int mapValue = ModalRoute.of(context).settings.arguments;
17     if(mapValue == null){
18       mapValue = 0;
19     }
20     Widget mapShowned = _getMapShowned(context, _height, _width, mapValue)↵
      ↵ ;
21
22     return Scaffold(
23       body: mapShowned
24     );
25   }
26
27   Widget _getMapShowned(BuildContext context, double height, double ↵
      ↵ width, int mapValue) {
28     switch (mapValue) {
29       case 1:
30         return AnastasyaTemplate.contentXL(context, height, width);
31         break;
32       case 2:
33         return ViccoTemplate.contentXL(context, height, width);

```

```
34     break;
35     case 3:
36         return HikariTemplate.contentXL(context, height, width);
37         break;
38     case 4:
39         return DeltaTemplate.contentXL(context, height, width);
40         break;
41     case 5:
42         return KeplerianTemplate.contentXL(context, height, width);
43         break;
44     case 0:
45     default:
46         return VladTemplate.contentXL(context, height, width);
47     }
48 }
49
50 }
```

Vista de pre-nivel

Código 3.17: Código .dart de vista de preniveles - IPT World

```

1 import 'package:flutter/material.dart';
2 import 'package:flutter/services.dart';
3
4 import 'package:ipt_world/src/components/atoms/colors.dart';
5 import 'package:ipt_world/src/components/templates/pre_level_template.↵
  ↵ dart';
6 import 'package:ipt_world/src/core/models/Station.dart';
7
8 class PreLevelPage extends StatefulWidget {
9
10  @override
11  _PreLevelPageState createState() => _PreLevelPageState();
12 }
13
14 class _PreLevelPageState extends State<PreLevelPage> {
15
16  @override
17  dispose() {
18    SystemChrome.setPreferredOrientations([DeviceOrientation.portraitUp↵
  ↵ ]);
19    super.dispose();
20  }
21
22  @override
23  Widget build(BuildContext context) {
24    SystemChrome.setPreferredOrientations([
25      DeviceOrientation.landscapeRight,
26      DeviceOrientation.landscapeLeft
27    ]);
28
29    final double _width = MediaQuery.of(context).size.width;
30    final double _height = MediaQuery.of(context).size.height;
31    Station station = ModalRoute.of(context).settings.arguments;
32
33    List<Color> colors = ColorEnum.getThemeArrayColors(station.appLevel)↵
  ↵ ;
34
35    return Scaffold(
36      backgroundColor: ColorEnum.white,
37      body: PreLevelTemplate.index(context, _height, _width, station, ↵
  ↵ colors)

```

```
38     );  
39 }  
40 }
```

Vista de pre-mapa

Código 3.18: Código .dart de vista de premapas - IPT World

```

1 import 'package:flutter/material.dart';
2
3 import 'package:ipt_world/src/utils/levelsTheme.dart' as theme;
4 import 'package:ipt_world/src/components/atoms/colors.dart';
5 import 'package:ipt_world/src/components/organisms/navigation_bar.dart'↵
    ↵';
6 import 'package:ipt_world/src/components/templates/level_info_template.↵
    ↵dart';
7 import 'package:ipt_world/src/core/models/Levels.dart';
8
9 class PreMapPage extends StatefulWidget {
10
11   @override
12   _PreMapPageState createState() => _PreMapPageState();
13 }
14
15 class _PreMapPageState extends State<PreMapPage> {
16
17   int _query;
18
19   @override
20   Widget build(BuildContext context) {
21
22     _query = ModalRoute.of(context).settings.arguments;
23     if(_query == null){
24       _query = 0;
25     }
26     final double _width = MediaQuery.of(context).size.width;
27     final double _height = MediaQuery.of(context).size.height;
28     List<Levels> levels = Levels.getMockLevels();
29     print(levels);
30
31     return Scaffold(
32       backgroundColor: ColorEnum.white,
33       body: LevelInfoTemplate.content(_height, _width, levels[_query], ↵
        ↵context),
34       bottomNavigationBar: NavigationBar.navigationBar(context, _height, ↵
        ↵_width, 1, theme.textColor(levels[_query].color, 0.6))
35     );
36   }
37 }

```

Vista de registro

Código 3.19: Código .dart de vista de registro - IPT World

```
1 import 'package:flutter/material.dart';
2
3 import 'package:ipt_world/src/components/templates/register_template.dart';
4
5 class RegisterPage extends StatelessWidget {
6
7   @override
8   Widget build(BuildContext context) {
9
10     final double _width = MediaQuery.of(context).size.width;
11     final double _height = MediaQuery.of(context).size.height;
12
13     return Scaffold(
14       body: RegisterTemplate.content(context, _height, _width),
15     );
16   }
17
18 }
```

3.5.4 Útiles

Levels Theme

Código 3.20: Código .dart del tema de niveles - IPT World

```
1 import 'package:flutter/material.dart';
2 import 'package:ipt_world/src/components/atoms/colors.dart';
3
4 Color textColor(String color, double opacity) {
5
6   switch (color) {
7     case 'vampire-color':
8       return ColorEnum.vampire(opacity);
9       break;
10    case 'anastasya-color':
11      return ColorEnum.anastasya(opacity);
12      break;
13    case 'vicco-color':
14      return ColorEnum.vicco(opacity);
15      break;
16    case 'hikari-color':
17      return ColorEnum.hikari(opacity);
18      break;
19    case 'delta-color':
20      return ColorEnum.delta(opacity);
21      break;
22    case 'keplerian-color':
23      return ColorEnum.keplerian(opacity);
24      break;
25    default:
26      return ColorEnum.black;
27      break;
28  }
29
30 }
```

Provider

Código 3.21: Código .dart de los proveedores de servicios - IPT World

```
1 import 'package:flutter/material.dart';
2 import 'package:ipt_world/src/core/blocs/user-bloc.dart';
3
4 class Provider extends InheritedWidget {
5   static Provider _instancia;
6   final _userBloc = new UserBloc();
7
8   factory Provider({Key key, Widget child}) {
9     if (_instancia == null)
10       _instancia = new Provider._internal(key: key, child: child);
11
12     return _instancia;
13   }
14
15   Provider._internal({Key key, Widget child}) : super(key: key, child: ↵
        ↵ child);
16
17   @override
18   bool updateShouldNotify(InheritedWidget oldWidget) => true;
19
20   static UserBloc of(BuildContext context) {
21     return context.dependOnInheritedWidgetOfExactType<Provider>().↵
        ↵ _userBloc;
22   }
23
24 }
```

Routes

Código 3.22: Código .dart de la lógica de rutas - IPT World

```
1 import 'package:flutter/material.dart';
2 import 'package:ipt_world/src/pages/activities/t-io_page.dart';
3 import 'package:ipt_world/src/pages/activities/t-ip_page.dart';
4 import 'package:ipt_world/src/pages/activities/t-oo_page.dart';
5 import 'package:ipt_world/src/pages/auth_page.dart';
6
7 import 'package:ipt_world/src/pages/home_page.dart';
8 import 'package:ipt_world/src/pages/map_page.dart';
9 import 'package:ipt_world/src/pages/pre_level_page.dart';
10 import 'package:ipt_world/src/pages/pre_map_page.dart';
11 import 'package:ipt_world/src/pages/register_page.dart';
12
13 Map<String, WidgetBuilder> getRoutes() {
14   return <String, WidgetBuilder>{
15     'home': (BuildContext context) => HomePage(),
16     'auth': (BuildContext context) => AuthPage(),
17     'register': (BuildContext context) => RegisterPage(),
18     'map': (BuildContext context) => MapPage(),
19     'pre-map': (BuildContext context) => PreMapPage(),
20     'pre-level': (BuildContext context) => PreLevelPage(),
21
22     // Activity types routes
23     'T-IP': (BuildContext context) => TipPage(),
24     'T-OO': (BuildContext context) => TooPage(),
25     'T-IO': (BuildContext context) => TioPage()
26   };
27 }
```

Validators

Código 3.23: Código .dart de validadores - IPT World

```

1 import 'dart:async';
2
3 class Validators {
4
5   final validateEmail = StreamTransformer<String, String>.fromHandlers(
6     handleData: ( email, sink ) {
7       Pattern pattern = r'^(([^<>() []\.\.,;:\s@"']+(\.[^<>() []\.\.,;:\s@
           ↳ \"]+)*)|(\\".+\"))@
           ↳ ((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\)|((\[a-
           ↳ zA-Z\-[0-9]+\.)+[a-zA-Z]{2,}))$';
8       RegExp regExp = new RegExp(pattern);
9
10      if(regExp.hasMatch(email))
11        sink.add( email );
12      else
13        sink.addError('Email no es correcto');
14    }
15  );
16
17 }

```

3.5.5 Retroalimentación del proyecto

El aplicativo de IPT World alcanza unos avances considerables antes de su respectivo paso a producción, se aplicó muy bien la tecnología utilizada y sobre todo las buenas prácticas de codificación y desarrollo de proyectos con equipos de trabajo. Este proyecto suma importancia en la empresa aunque no sea el proyecto con más prioridad, permite a los estudiantes de la academia contar con una aplicación móvil que les permite aprender mediante la diversión y el entretenimiento, cabe recalcar que estará sujeta a cambios, actualizaciones y posibles regresos de producción.

Se manejó muy bien el control de versiones por parte del equipo de trabajo, haciendo mucho énfasis en las buenas prácticas y manejo de Git y GitHub, así mismo se continuó con la ruta de aprendizaje prolongada en Flutter y programación orientada a objetos por parte de todo el equipo de trabajo, finalmente el proyecto se presentó al product owner y los resultados fueron de satisfacción.

3.6 IPT Platform

La plataforma IPT es el proyecto más importante de la empresa por su gran requerimiento en cada uno de los departamentos, la unificación de todas las funcionalidades y lógicas de negocio le permitirían a la academia optimizar todas las funciones que se realizan convencionalmente.

3.6.1 Requerimientos de la plataforma

Se necesita una plataforma web capaz de dividir cada dependencia de la empresa y procesar flujos de negocio en función al departamento seleccionado, principalmente el departamento académico.

Dentro de los requisitos principales se encuentra la creación de usuarios por parte de los coordinadores de cada dependencia, para darle un rol determinado a cada empleado de la empresa y este pueda desarrollar digitalmente las actividades que realiza de manera manual y convencional.

Los requerimientos de la plataforma son presentados por parte de cada coordinador de departamento y principalmente por los jefes de la empresa, se exponen de manera clara al equipo de desarrollo las funciones que se esperan al finalizar una primera versión del aplicativo, con ello se presenta un flujo y estructura de lo que será el primer release de la plataforma web, teniendo como objetivos específicos la categorización del personal de la empresa y explicación de sus roles, adicionalmente retroalimentar los requerimientos con una conceptualización de las funciones de cada dependencia presentadas con historias de usuario para planificación de tareas. Finalmente y no menos importante la estimación de tiempos por parte del equipo de desarrollo es crucial para mantener al personal informado del desarrollo y nivel del equipo en general, claramente las estimaciones se realizan con mucho análisis y cuidado para tener una buena precisión.

3.6.2 Historias de usuario y metodología

Como pilares de desarrollo se necesita mucha transparencia, inspección y adaptación por el equipo, optando nuevamente por metodología scrum manteniendo una jerarquía en el mismo, tales como products owner, scrum master, equipo de desarrollo general e hitos principales. Estos hitos son los siguientes:

- Daily: Son encuentros online diarios realizados por el equipo de desarrollo para hacer reportes breves de lo trabajado cada 24 horas.
 - Checkstatus: Son reuniones realizadas entre el equipo de desarrollo y los product owners para presentar los avances finales de cada Sprint
 - Sprint planning: Son reuniones realizadas solamente por el equipo de desarrollo para planificar actividades y estimación del siguiente Sprint.
 - Sprint: Son rutas de desarrollo implementadas cada 15 días en las que se abarcan una o más historias de usuario, generalmente un Sprint tiene una duración de dos semanas, la cual el último día tiende a ser el checkstatus.
-

Historias de usuario - IPT Platform

Particularmente este proyecto contiene muchas historias de usuario, por esa razón se ideó una ruta crítica por el equipo de desarrollo para organizar esa estructura relacionada al release 1.0, sin embargo se siguen manteniendo las HU's clasificadas por departamento. A continuación se presentan las historias de usuario más relevantes por departamento.

- Departamento académico

HU 003	
COMO	Usuario: Jefe Académico.
QUIERO	Tener una sección para crear y modificar usuarios profesores
PARA	Generar usuarios de profesores con los siguientes datos: - Nombres - Apellidos - Sexo - Tipo de documento - Número de documento - Ciudad de residencia - Teléfono - Correo electrónico - Fecha de nacimiento - Horas de servicio - Nickname - Contraseña Borrar usuarios de profesores inactivos Modificar cualquier dato el usuario PRO
PRIORIDAD	Muy alta
PUNTAJE	8

Figura 3.39: Historia de usuario - creación de usuario - Departamento académico.

Fuente: Autor

- Departamento administrativo

HU 009	
COMO	Usuario Admin Administrativo
QUIERO	Visualizar el seguimiento de pagos del departamento de cartera
PARA	Vigilar el comportamiento de pago de los estudiantes, conocer su progreso para generar procesos administrativos tales como finalización de estudios y constancia de estudios
PRIORIDAD	Muy alta
PUNTAJE	5

Figura 3.40: Historia de usuario - seguimiento de pagos - Departamento administrativo.

Fuente: Autor

- Departamento comercial

HU 015	
COMO	Usuario Asesor comercial
QUIERO	Generar el COD en la plataforma y poder crear los usuarios estudiantes en la plataforma
PARA	El registro de los estudiantes de nuevo ingreso con las respectivas validaciones dependiendo si es IPT Kids y de serlo registrar los datos adicionales del acudiente. El registro de los estudiantes de nuevo ingreso el cual va a ser diligenciado por los asesores comerciales, este COD cambiará de acuerdo al tipo de matrícula que se realice ya sea • IPT Kids: -Logo IPT Kids -Fotos: En la parte superior del COD se deben adjuntar las siguientes fotos: Foto acudiente 1, foto acudiente 2, foto estudiante. - Observaciones: Campo de texto en donde el asesor coloca algún dato o información que no se contemple en el COD, tales como permisos para llegar tarde a clase entre otros -Perfil Médico: Sección en la cual se tienen en cuenta los siguientes campos de texto: Medicamentos, enfermedades, alergias, PMR, RH

Figura 3.41: Historia de usuario - generación de COD - Departamento comercial.

Fuente: Autor

- Departamento de marketing

HU 014	
COMO	Usuario Admin MKT/COM
QUIERO	Tener una sección de los datos de los estudiantes que realizaron el placement test (ya sea universitario, colegio o sin convenio) para poder hacer marketing a través del celular o el correo Los datos serán los siguientes: • Nombres y Apellidos • Edad • Número de documento • Ciudad • Correo electrónico • Tipo de convenio • Puntaje y Nivel obtenido • Teléfono
PARA	Que el formulario cuente con los campos y la información suficiente para facilitar la labor del asesor comercial. Que el director de marketing pueda acceder a la información fácilmente y redireccionarla a los respectivos asesores comerciales.
PRIORIDAD	Alta
PUNTAJE	5

Figura 3.42: Historia de usuario - registro de Placement Test - Departamento de marketing.

Fuente: Autor

3.6.3 Presentación del Release 1.0

Release 1.0 hace referencia a la primera versión de la plataforma, teniendo la culminación de las historias de usuario planteadas en la primera entrada a producción y respetando su alcance, es decir, sin modificar las actividades a desarrollar a lo largo de la ejecución del proyecto. La presentación se realiza días previos al inicio del desarrollo, entregando a los propietarios de producto el alcance real que se tendrá con fechas estimadas netamente por el equipo.

Algo importante a tener en cuenta son los roles de cada departamento, y es algo que todo el sistema de la empresa debe tener claro antes de hacer uso de la plataforma y todas sus funcionalidades, por ello, se hace la presentación de los roles por medio del siguiente diagrama:

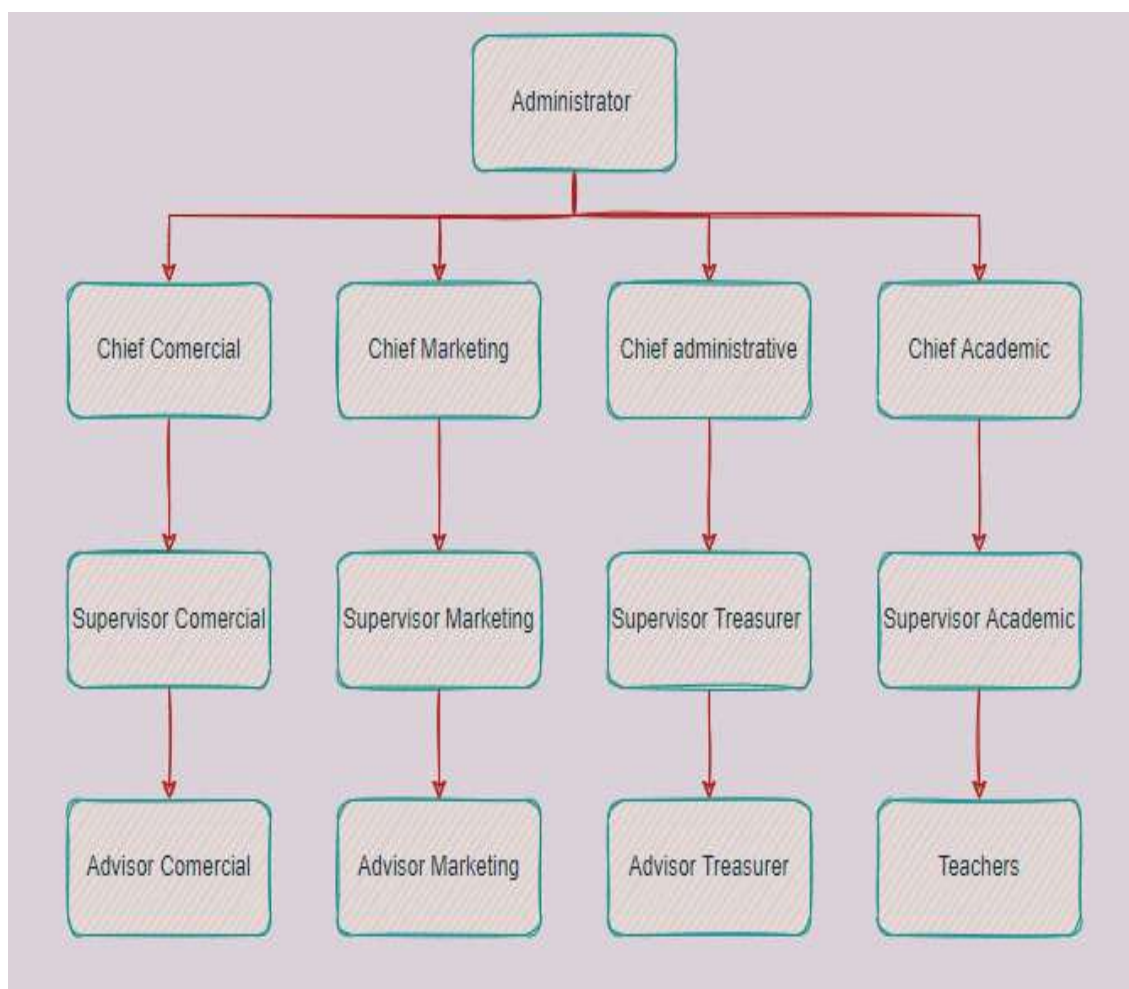


Figura 3.43: Roles dentro de la plataforma IPT.

Fuente: Autor

Dentro del sistema relacionado a la base de datos, los roles están clasificados por un administrador, el cual serán credenciales que usarán los jefes y los desarrolladores, los roles relacionados a 'Chief' serán para los coordinadores de cada departamento, en cuanto a los 'Supervisor' serán esos usuarios que tengan menos permisos que los Chief pero más permisos que los 'Advisor', que finalmente serán el resto de usuarios con permisos básicos en cada dependencia. Cabe recalcar que dentro de la página el sistema de roles tendrán nombres diferentes para cumplir con experiencia de usuario.

Ruta Crítica Plataforma IPT

Realizado por: Leiner Barrios - Alejandro Quiñones

Departamento de sistemas IPT

Prioridad

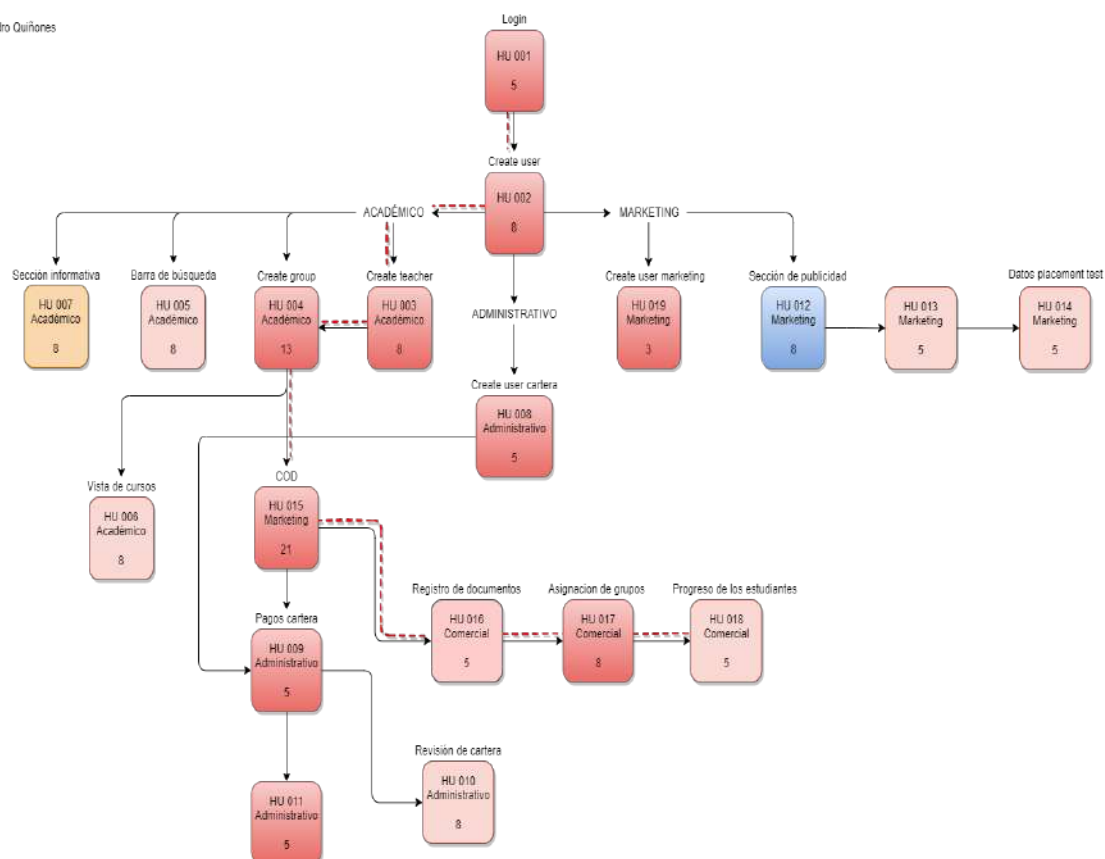


Figura 3.44: Diagrama general del Release 1.0 con ruta crítica.

Fuente: Autor

3.6.4 Desarrollo del proyecto

Continuando con la metodología atomic design, se mantienen la misma estructura para los componentes que conforman cada vista.

Sprint 1

Conformado por diferentes secciones, principalmente una vista principal de la página, générica y representativa para todas las dependencias, conteniendo en sí, secciones de publicidad, información y algunos redireccionamientos a diferentes páginas de la empresa.

- Vista principal: Fue en gran porcentaje desarrollada por el practicante, realizando una maquetación puramente con HTML, aplicando estilos desde cero con CSS y sin el uso de frameworks, las principales propiedades aplicadas fueron CSS Grid y FlexBox.

Código 3.24: Código TS de vista de Index - Plataforma IPT

```
1 import { Component, EventEmitter, Input, OnInit, Output } from '@angular/core';
2 import { UserRs } from 'src/app/core/models/user/user-rs.model';
3
4 @Component({
5   selector: 'app-index-template',
6   templateUrl: './index-template.component.html',
7   styleUrls: ['./index-template.component.css'],
8 })
9 export class IndexTemplateComponent implements OnInit {
10   @Input() copies: any;
11   @Output() register = new EventEmitter<boolean>();
12   @Input() validatorRoutes!: boolean;
13   @Output() home = new EventEmitter<boolean>();
14   @Output() formType = new EventEmitter<number>();
15   @Output() pageChange = new EventEmitter<string>();
16
17   user!: UserRs;
18
19   constructor() {
20   }
21
22   clickRegister() {
23     var form = document.createElement('form');
24     document.body.appendChild(form);
25     form.method = 'get';
26     form.action = 'http://localhost:4201/';
```

```
27     form.submit();
28   }
29
30
31   ngOnInit(): void {
32   }
33
34   clickedLink(event: number) {
35     this.formType.emit(event);
36   }
37 }
```

- Login: Contiene parte de algo que se desarrolló anteriormente, luego de tener un trabajo hecho en EKT Platform, fue relativamente menos complicado realizar un login, este login contiene nuevos atomos, con diferentes estilos desarrollados por el aprendiz, así mismo contiene a su izquierda un carrusel de imágenes que podrían redireccionar a sección publicitaria e informativa, cabe aclarar que el diseño del login fue desarrollado por el equipo de marketing, entregándole al departamento de sistemas, una plantilla de como se espera el resultado de la vista de login, a continuación se presenta el código que contiene la lógica detrás de toda la autenticación.

Código 3.25: Código TS de vista de Login - Plataforma IPT

```
1 export class LoginComponent {
2
3   @Input () copies: any;
4   @Input() validatorRoutes!: boolean;
5   @Output() login = new EventEmitter<boolean>();
6   @Output() formType = new EventEmitter<number>();
7   @Output() pageChange = new EventEmitter<string>();
8
9   pageOption: string = '';
10  loginForm!: FormGroup;
11  loader: boolean = false;
12  user!: UserRs;
13
14  constructor(
15    fb: FormBuilder,
16    private userService: UserService,
17    private storageService: StorageService ) {
18    this.loginForm = fb.group({
19      email: ['', [Validators.required, Validators.maxLength(100)]],
20      password: ['', [Validators.required, Validators.minLength(4), ↵
                ↵ Validators.maxLength(32)]],
21    });
22  }
23
24  clickLogin() {
25    var form = document.createElement('form');
26    document.body.appendChild(form);
27    form.method = 'get';
28    // form.target = '_blank';
29    form.action = "http://localhost:4202/";
30    form.submit();
31  }
32}
```

- Register: En esta vista simplemente se realiza un cambio de vista de la columna derecha, se mantiene ese carrusel de imágenes y ahora parecen muchas más entradas de registro, cumpliendo con los parámetros de datos requeridos para un registro.

Código 3.26: Código TS de Register - Plataforma IPT

```

1 export class RegisterComponent {
2
3   @Input() copies: any;
4   @Output() register = new EventEmitter<boolean>();
5   @Output() formType = new EventEmitter<number>();
6   @Output() sessionChange = new EventEmitter<boolean>();
7   @Output() pageChange = new EventEmitter<string>();
8
9   registerForm!: FormGroup;
10  loader: boolean = false;
11  user!: UserRs;
12  idTypes: any;
13  indicatives: any;
14
15  constructor(
16    private parameterService: ParameterService,
17    fb: FormBuilder,
18    private userService: UserService,
19    private storageService: StorageService
20  ) {
21    this.registerForm = fb.group({
22      name: ['', [Validators.required, Validators.minLength(3), ↵
23        ↵ Validators.maxLength(32)]],
24      last_name: ['', [ Validators.required, Validators.minLength(5), ↵
25        ↵ Validators.maxLength(32)]],
26      id_card_type: ['', [Validators.required]],
27      id_card: ['', [Validators.required, Validators.minLength(5), ↵
28        ↵ Validators.maxLength(20)]],
29      indicative: ['', [Validators.required]],
30      phone_number: ['', [Validators.required, Validators.minLength(↵
31        ↵ (10), Validators.maxLength(15)]],
32      email: ['', [Validators.required, Validators.maxLength(300), ↵
33        ↵ this.dominio]],
34      password: ['', [Validators.required, Validators.minLength(7), ↵
35        ↵ Validators.maxLength(32)]],
36      re_password: ['', [Validators.required, Validators.minLength(7) ↵
37        ↵ , Validators.maxLength(32)]],
38    });

```

```
32 }  
33 }
```

- Forgot-password: La sección de olvidar contraseña es mucho más corta, y simplemente se agregan los campos del email y número de documento, con el fin de validar a la persona con esa identificación y solicitar su correo para el envío de la nueva contraseña.

Código 3.27: Código TS de Forgot-password - Plataforma IPT

```
1 export class ForgotPasswordComponent {
2
3   @Input () copies: any;
4   @Output() forgot = new EventEmitter<boolean>();
5   @Output() formType = new EventEmitter<number>();
6   @Output() pageChange = new EventEmitter<string>();
7
8   forgotForm!: FormGroup;
9   loader: boolean = false;
10
11  constructor(
12    fb: FormBuilder) {
13    this.forgotForm = fb.group({
14      email: ['', [Validators.required, Validators.maxLength(100), ↵
15        ↵ this.dominio]],
16      idCard: ['', [Validators.required, Validators.minLength(5), ↵
17        ↵ Validators.maxLength(20)]],
18    });
19  }
20
21  dominio(control: AbstractControl): ValidationErrors | null {
22    let dominio: string = control.value;
23    let size = dominio.split('@inglesparatodos.edu.co').length;
24    if (size === 2) {
25      return null;
26    } else {
27      return { dominio: true }
28    }
29  }
30
31  carouselAction() {
32    this.pageChange.emit('announcement');
33  }
34
35  ngOnInit(): void {
36  }
```

- Dashboard principal: El dashboard principal se encuentra dentro del Sprint 1, en el cual se desarrolló principalmente la creación de usuarios para cada departamento, es un proceso repetitivo pero que con el primer desarrollo se pueden replicar los demás, es decir, realizar una copia para luego poder cambiar nombres y representaciones; esto para repositorio los cuales hacen referencias a algo que se llama micro-frontends.

Código 3.28: Código TS de dashboard principal - Plataforma IPT

```
1 import { Component, EventEmitter, Input, OnInit, Output } from '@angular/core';
2
3 @Component({
4   selector: 'app-dashboard-page',
5   templateUrl: './dashboard-page.component.html',
6   styleUrls: ['./dashboard-page.component.css']
7 })
8 export class DashboardPageComponent {
9
10  @Input () copies: any;
11  @Input() validatorRoutes!: boolean;
12  @Input() blockTimer!: boolean;
13  @Output() login = new EventEmitter<boolean>();
14  @Output() formType = new EventEmitter<number>();
15  @Output() sessionChange = new EventEmitter<boolean>();
16  @Output() pageChange = new EventEmitter<string>();
17
18  pageOption: string = 'home';
19  setPage: string = 'index';
20
21  constructor() { }
22
23  navBarAction(event: string) {
24    // this.pageChange.emit(event);
25    this.pageOption = event;
26  }
27
28  isClickedLogout(){
29    this.setPage = 'logout';
30    this.navBarAction('logout');
31  }
32
33  changePage(event: string) {
34    this.pageOption = event;
35  }
```

```
36  
37 sessionChangeAction(){  
38     this.sessionChange.emit(true);  
39 }  
40  
41 }
```

Actividades faltantes del Sprint 1:

- Creación de usuario académico
- Creación de usuario administrativo
- Creación de usuario de marketing
- Creación de usuario comercial

Sprint 2

Dentro del sprint 2 se encuentran las mismas funcionalidades del sprint 1, particularmente este sprint tuvo un desarrollo relativamente corto en comparación con el primero, debido a que la funcionalidad en general debe ser tendencia en cada dependencia, por lo tanto elaborar una vista de creación de usuarios igual para todos los administradores es fundamental y útil para mantener estandarizada la plataforma y la experiencia de usuario, así mismo se están aplicando buenas prácticas para la escritura de código ya que se están reutilizando los mismo métodos, estilos, lógica y servicios de back. Como pocas diferencias se pueden encontrar en los títulos, imágenes, vistas por departamento y algunos detalles más, sin embargo se mantienen los mismo campos de registro y de inserción de datos.

Sprint 3

Este sprint está conformado por dos importantes historias de usuario, las cuales son considerablemente extensas, al ser así se realiza una división de las mismas, por lo que al aprendiz le correspondió todo lo relaciones a cursos (Creación de cursos, edición de cursos, creación de copia de los cursos, vista general de los cursos, vista de los cursos activos por nivel.)

- Creación de cursos: La lógica para la creación de cursos fue un reto para el aprendiz, la realización del mismo dependa de la continuación de las demás funcionalidades que respectan a los cursos, primeramente debe realizarse un Look and feel, es ese diseño o maquetación que se realiza previamente, posterior a eso se realiza la funcionalidad, es decir, la validación tanto del formulario como de botones, y finalmente se aplica la integración con los servicios de la API, este proceso suele ser el más complicado de todos, ya que lleva mucho análisis y siempre se suelen presentar muchos errores y bugs.

Código 3.29: Código TS de creación de curso académico - Plataforma IPT

```
1 export class CreateCourseModalComponent implements OnInit {
2   @Input() copies: any;
3   @Input() validatorRoutes!: boolean;
4   @Output() formType = new EventEmitter<number>();
5   @Output() pageChange = new EventEmitter<string>();
6
7   createCourseForm!: FormGroup;
8   loader: boolean = false;
9   user!: UserRs;
10  levels: any;
11  agreements: any;
12  coursesTypes: any;
13  classrooms: any;
14
15  academicPlan: any[] = [
16    { code: '1', name: 'Academy' },
17    { code: '2', name: 'Social Kids' },
18    { code: '3', name: 'Universities' },
19    { code: '4', name: 'Business' },
20  ];
21
22  ngOnInit(): void {
23    this.parameterService.getCoursesTypes().subscribe((response) => {
24      this.coursesTypes = response.data;
25    });
26    this.parameterService.getLevels().subscribe((response) => {
27      this.levels = response.data;
```

```

28     });
29     this.parameterService.getAgreements().subscribe((response) => {
30         this.agreements = response.data;
31     });
32     this.parameterService.getAllClassrooms().subscribe((response) => ↵
33         ↵ {
34         this.classrooms = response.data;
35     });
36 }
37 constructor(
38     fb: FormBuilder,
39     private userService: UserService,
40     private parameterService: ParameterService
41 ) {
42     this.createCourseForm = fb.group({
43         start_date: ['', [Validators.required]],
44         finish_date: ['', [Validators.required]],
45         time_start_monday: ['', [Validators.required]],
46         time_finish_monday: ['', [Validators.required]],
47         time_start_tuesday: ['', [Validators.required]],
48         time_finish_tuesday: ['', [Validators.required]],
49         time_start_wednesday: ['', [Validators.required]],
50         time_finish_wednesday: ['', [Validators.required]],
51         time_start_thursday: ['', [Validators.required]],
52         time_finish_thursday: ['', [Validators.required]],
53         time_start_friday: ['', [Validators.required]],
54         time_finsih_friday: ['', [Validators.required]],
55         time_start_saturday: ['', [Validators.required]],
56         time_finish_saturday: ['', [Validators.required]],
57         levels: ['', [Validators.required]],
58         coursesTypes: ['', [Validators.required]],
59         academic_plan: ['', [Validators.required]],
60         agreement: ['', [Validators.required]],
61         classroom: ['', [Validators.required]],
62         capacity: ['', [Validators.required, Validators.minLength(1), ↵
63             ↵ Validators.maxLength(2)]],
64         more_days: ['', [Validators.required]],
65         teachers: ['', [Validators.required]],
66     });
67 }
68
69 actionButton() {
70     this.loader = true;

```

```
71 let request: CreateCourseRq = {
72   academic_plan: this.createCourseForm.controls['academic_plan'].↵
       ↵ value,
73   coursesTypes: this.createCourseForm.controls['coursesTypes'].↵
       ↵ value,
74   classroom: this.createCourseForm.controls['classroom'].value,
75   levels: this.createCourseForm.controls['levels'].value,
76   agreement: this.createCourseForm.controls['agreement'].value,
77   capacity: this.createCourseForm.controls['capacity'].value,
78   teachers: this.createCourseForm.controls['teachers'],
79   start_date: this.createCourseForm.controls['start_date'].value,
80   finish_date: this.createCourseForm.controls['finish_date'].↵
       ↵ value,
81   time_start_monday: this.createCourseForm.controls['↵
       ↵ time_start_monday'].value,
82   time_finish_monday: this.createCourseForm.controls['↵
       ↵ time_start_monday'].value,
83   time_start_tuesday: this.createCourseForm.controls['↵
       ↵ time_start_monday'].value,
84   time_finish_tuesday: this.createCourseForm.controls['↵
       ↵ time_start_monday'].value,
85   time_start_wednesday: this.createCourseForm.controls['↵
       ↵ time_start_monday'].value,
86   time_finish_wednesday: this.createCourseForm.controls['↵
       ↵ time_start_monday'].value,
87   time_start_thursday: this.createCourseForm.controls['↵
       ↵ time_start_monday'].value,
88   time_finish_thursday: this.createCourseForm.controls['↵
       ↵ time_start_monday'].value,
89   time_start_friday: this.createCourseForm.controls['↵
       ↵ time_start_monday'].value,
90   time_finsih_friday: this.createCourseForm.controls['↵
       ↵ time_start_monday'].value,
91   time_start_saturday: this.createCourseForm.controls['↵
       ↵ time_start_monday'].value,
92   time_finish_saturday: this.createCourseForm.controls['↵
       ↵ time_start_monday'].value,
93 };
94 }
```

- Vista de cursos: Luego de un formulario de creación de cursos, se necesita de una vista en la que se puedan visualizar esos cursos creados, el aprendiz fue capaz de implementar una tabla de Bootstrap, así como lo hizo en la plataforma de EKT, en este caso se necesitaron más campos y estilos diferentes, finalmente se pudo cumplir con la validación de que al crear un curso, se pueda visualizar en esa tabla.
 - Edición de cursos: El aprendiz desarrolló una lógica para la edición de los cursos, consistía primeramente en la misma que muchos formularios tienen, y es la realización de modales netamente con Angular y comunicación de componentes, se le aplicó una propiedad de (click) a cada fila de tabla, la cual pudiese detectar el code de ese curso y desplegarlo trayendo con él los campos ya pintados que tenía ese curso, fue una gran implementación, ya que se utilizaron tanto @Input() como @Output() para la comunicación de diferentes componentes o diferentes padres e hijos.
 - Copia de cursos: Este desarrollo tiene semejanzas con la edición de cursos, acá se desarrolló por parte de practicante una comunicación entre componentes mucho más larga, ya que primeramente se debía comunicar el modal de editar curso con el padre que vendría siendo el modal, luego otra comunicación entre el modal y la tabla, ya que al estar conectados se necesitaba llevar esa información del curso, luego de llegar a la tabla se le pasaban esos datos a la vista principal de cursos. Finalmente esa información era descendida nuevamente al componente de modal, ya que este se encargaba de contener todos los formularios, por lo tanto era importante que la información pasara por él para que luego fuese llevada al modal de crear curso, toda esta lógica fue ideada por el equipo de desarrolló y principalmente ejecutada por el aprendiz.
-

3.6.5 Avances entregados

Se ha desarrollado una gran parte del primer release, cumplimiento con las entregas en cada Sprint Review y con ello presentando avances considerables y de buena calidad, específicamente se contemplan las siguientes entregas:

- **Página principal:** Se hace entrega de una página principal conformada por varias secciones en las cuales se encuentran formulario de registros para promociones e información de cursos, sección informativa acerca de los cursos, link de redirección al Placement Test, información respectiva a la ubicación de la empresa y detalles varios.
 - **Página de ingreso:** Se entrega un vista singular de login, register y olvido de contraseña; en la cual se encuentra a un lado un scroll de imágenes cliqueables e informativas, esta página de ingreso es genérica tanto para los departamentos de la empresa como para los estudiantes.
 - **Dashboard de cada departamento:** Se ha simplificado el diseño del dashboard principal de cada dependencia, teniendo como únicas diferencias los títulos del panel administrador y botones del sidebar que dirigen a las diferentes secciones de determinado departamento.
 - **Creación de usuarios en cada departamento:** Se hace entrega general de creación de usuarios para cada dependencia, representado en un modal pequeño que permite crear rápidamente cualquier usuario con su determinado rol.
 - **Creación de cursos académicos:** Se ha probado y entregado una de las principales funcionalidades académicas como lo es la creación de grupos de clases, los cuales están conformados por un horario con su salón de clases o sala de conferencia para el caso de cursos virtuales, se permite la asignación de plan académico, tipo de plan, convenio en caso tal exista, nivel de inglés, profesor y algunos filtros más, se entrega un registro en base de datos de esta creación de cursos, así como la visualización del listado dividido por niveles.
-

3.6.6 Retroalimentación y continuidad del proyecto

Al ser un proyecto que se mantendrá siempre en desarrollo, las primeras entregas apuntan a un sólido recurso informático utilizado por la empresa, teniendo diversas tecnologías para soportar y contemplar las necesidades de cada departamento, es cierto que se han pedido muchas implementaciones, cambios y actualizaciones en contadas historias de usuario y esto es debido al gran impacto que han tenido los primeros avances. No obstante, es importante aclarar la importancia de las buenas prácticas de desarrollo respetando las primeras peticiones y requerimientos sin aplicar cambios significativos a lo largo del desarrollo del primer release 1.0.

La continuidad es un hecho, se partió desde requerimientos claros y bien definidos para dar entrega a muchas versiones futuras de la plataforma, en la que los cambios e implementaciones irán aumentando a medida que se requieran más funcionalidades internas de la empresa, y como objetivo principal, se busca la optimización de recursos académicos, administrativos, mejora en los procesos educativos e innovación de las propuestas de metodologías de enseñanza y respuesta eficiente del aprendizaje por parte de los estudiantes de la academia.

Ha sido notorio el aumento de transferencia de conocimiento en el equipo de desarrollo, así como la buena calidad del proyecto respecto al código limpio, buenas prácticas por parte de cada desarrollador y gran crecimiento del aprendiz, con esto, es relevante el hecho de poder contar en los futuros requerimientos de la empresa en cuanto a tecnología y desarrollo se refiere.

4 Resultados

4.1 EKT Platform

4.1.1 Vistas de Index

Las vistas de index son aquellas que conforman un vista previa para el usuario, esta vista le permite navegar entre componentes antes de dar uso general de la interfaz interna y principal de la página, en ella podemos encontrar vista de ingreso para usuarios creados, vista de registro para usuarios nuevos y vista de olvido de contraseña para ese caso.

Login

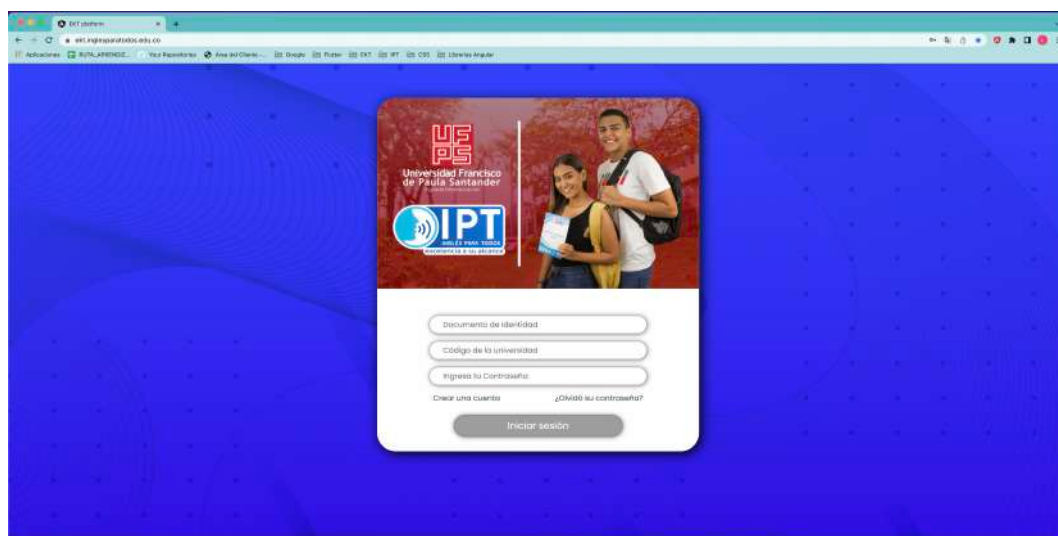


Figura 4.1: Vista de login - Plataforma EKT.

Fuente: Autor

Register

Figura 4.2: Vista de registro - Plataforma EKT.

Fuente: Autor

Forgot-password

Figura 4.3: Vista de olvidar contraseña - Plataforma EKT.

Fuente: Autor

4.1.2 Vistas del dashboard

En esta sección encontramos muchos más componentes, es la vista principal en la que administrador y el estudiante interactúa con la página en general, cabe recalcar que dentro de la lógica de esta vista, se tienen ciertas limitaciones para el estudiante como la vista de grupos y puntajes en tiempo real.

Previsualización de convocatorias

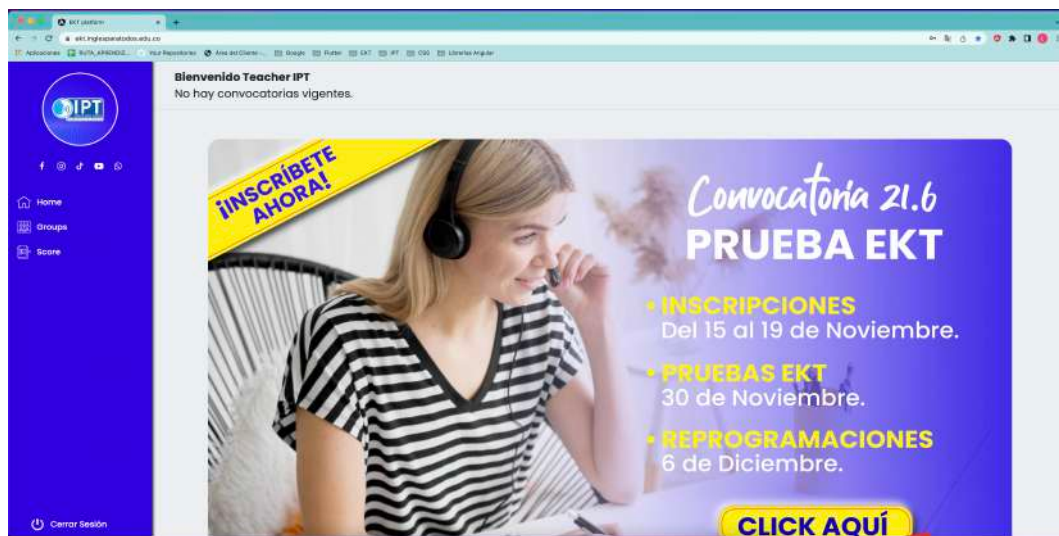


Figura 4.4: Vista de convocatorias vigentes y vencidas - Plataforma EKT.

Fuente: Autor

Video tutorial

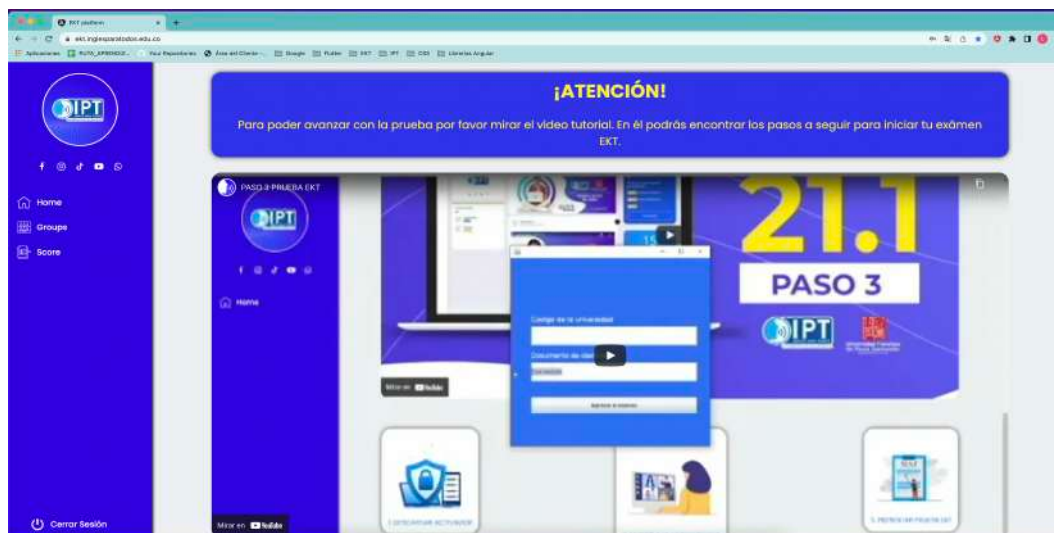


Figura 4.5: Vista de instrucción y video tutorial - Plataforma EKT.

Fuente: Autor

Pasos para presentar la prueba

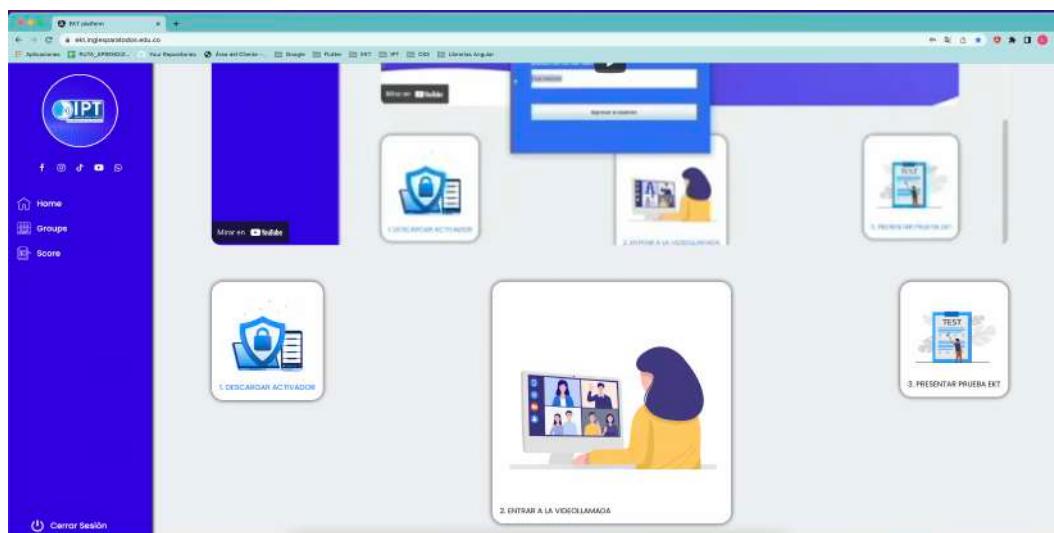


Figura 4.6: Vista de paso a paso para presentar la prueba EKT - Plataforma EKT.

Fuente: Autor

Vista de grupos de prueba

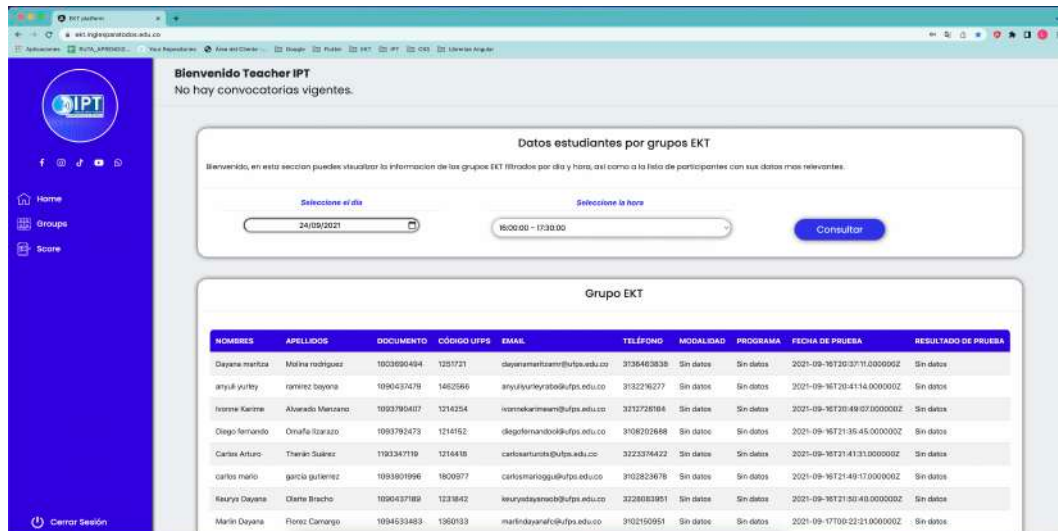


Figura 4.7: Vista de grupos de estudiantes para presentar prueba divididos por hora - Plataforma EKT.

Fuente: Autor

Vista de grupos con puntajes de prueba

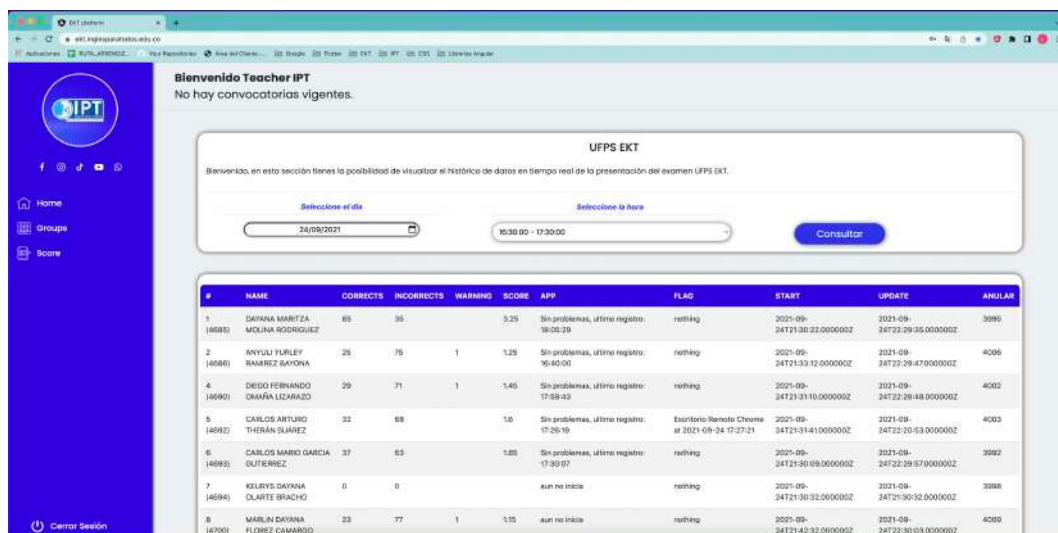


Figura 4.8: Vista de listado de estudiantes con puntaje - Plataforma EKT.

Fuente: Autor

4.2 IPT World

4.2.1 Vistas de login

En esta sección se presentan los avances equivalentes a la vista de ingreso, con una imagen vertical deslizable conformada por una portada, por una sección de ingreso y registro.

Portada de login



Figura 4.9: Vista de portada para login - IPT World.

Fuente: Autor

Login

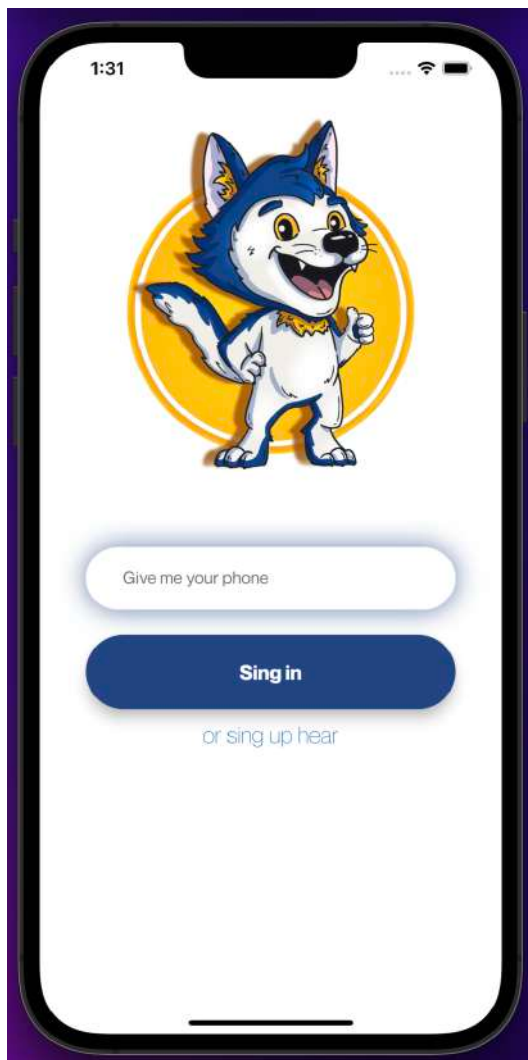


Figura 4.10: Vista de ingreso - IPT World.

Fuente: Autor

Register

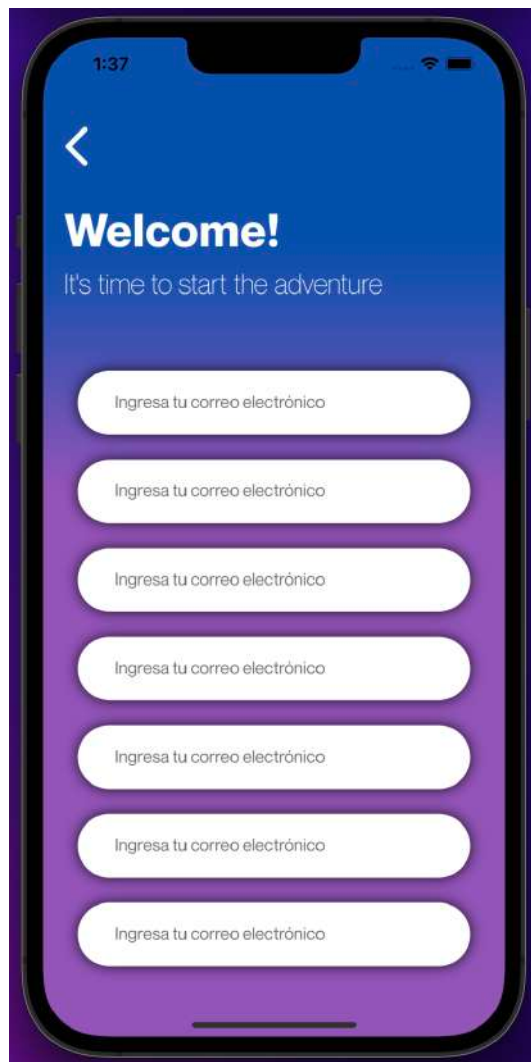


Figura 4.11: Vista de registro - IPT World.

Fuente: Autor

4.2.2 Vista principal con niveles

En este apartado se expone una ventana principal alcanzada, con niveles divididos y vistas individuales de cada uno, así como la ruta de mundos y su didáctica.

Niveles



Figura 4.12: Vista principales con lista horizontal de niveles - IPT World.

Fuente: Autor

Ingreso a mundos por niveles



Figura 4.13: Vista inicio a mundos con avatares de cada nivel - IPT World.

Fuente: Autor

4.3 IPT Platform

4.3.1 Página de index

La página de index es la principal, genérica y adaptable para todos las dependencias y público en general.



Figura 4.14: Vista de Index - IPT Platform.

Fuente: Autor

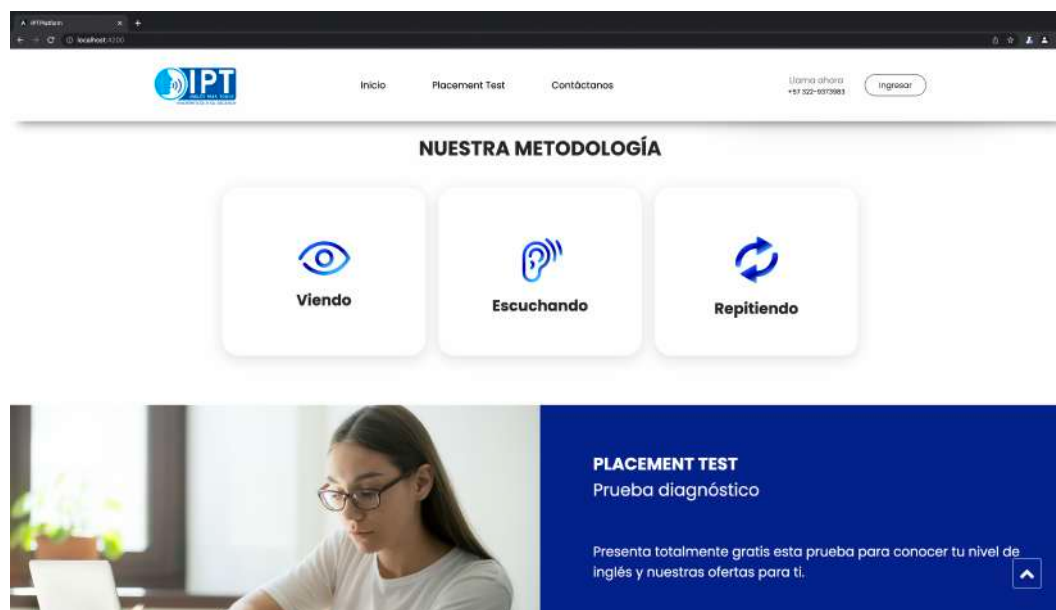


Figura 4.15: Vista de Index - IPT Platform.

Fuente: Autor

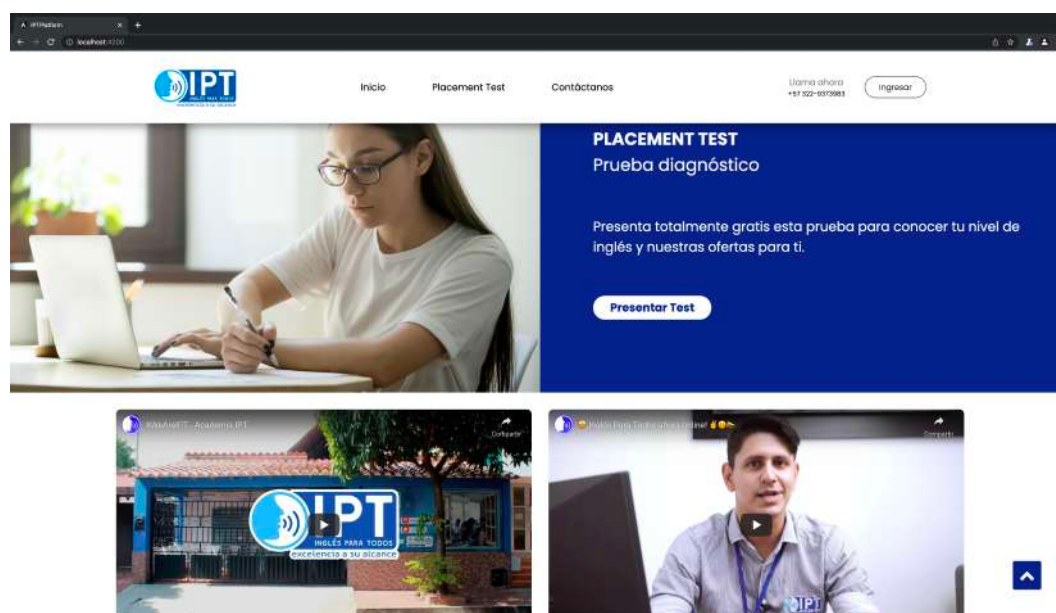


Figura 4.16: Vista de Index - IPT Platform.

Fuente: Autor

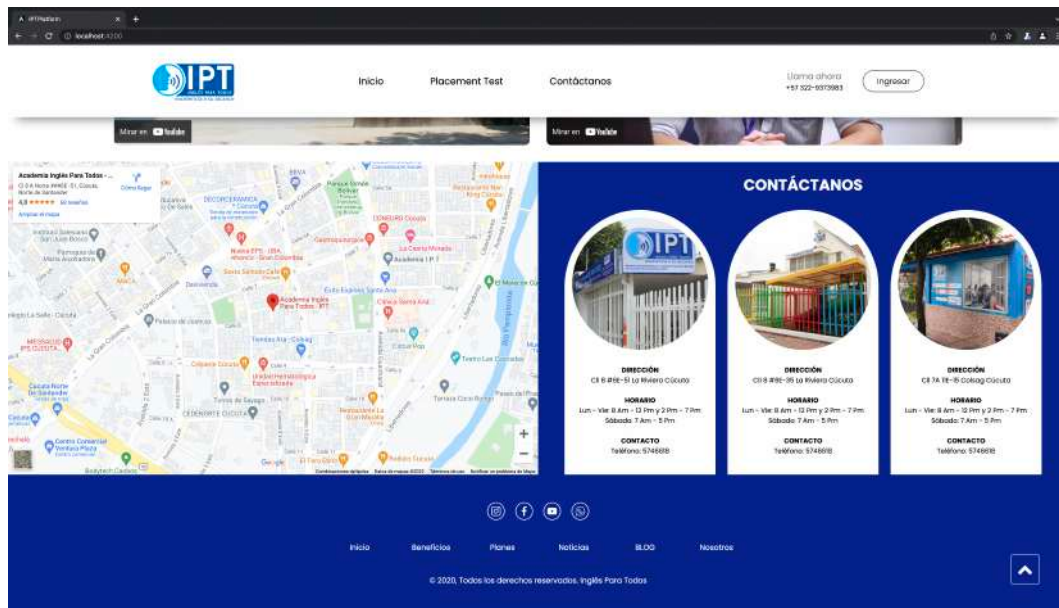


Figura 4.17: Vista de Index - IPT Platform.

Fuente: Autor

4.3.2 Ingreso

La página de ingreso está conformada por un login, register y forgot-password, acompañadas de un scroll de imágenes y esta es general para los miembros de la empresa, tanto empleados como estudiantes.

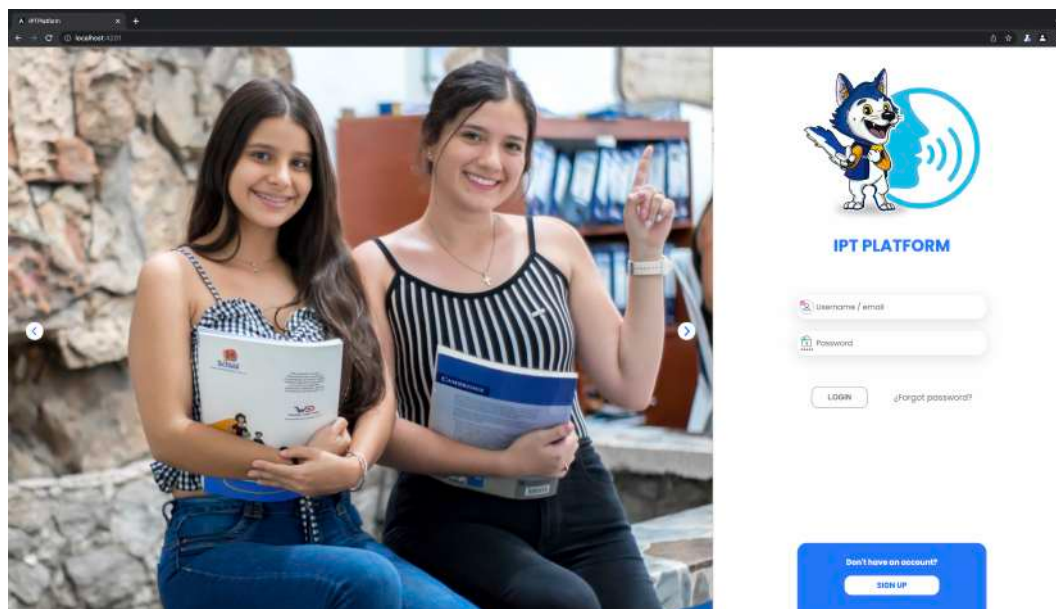


Figura 4.18: Vista de Login - IPT Platform.

Fuente: Autor

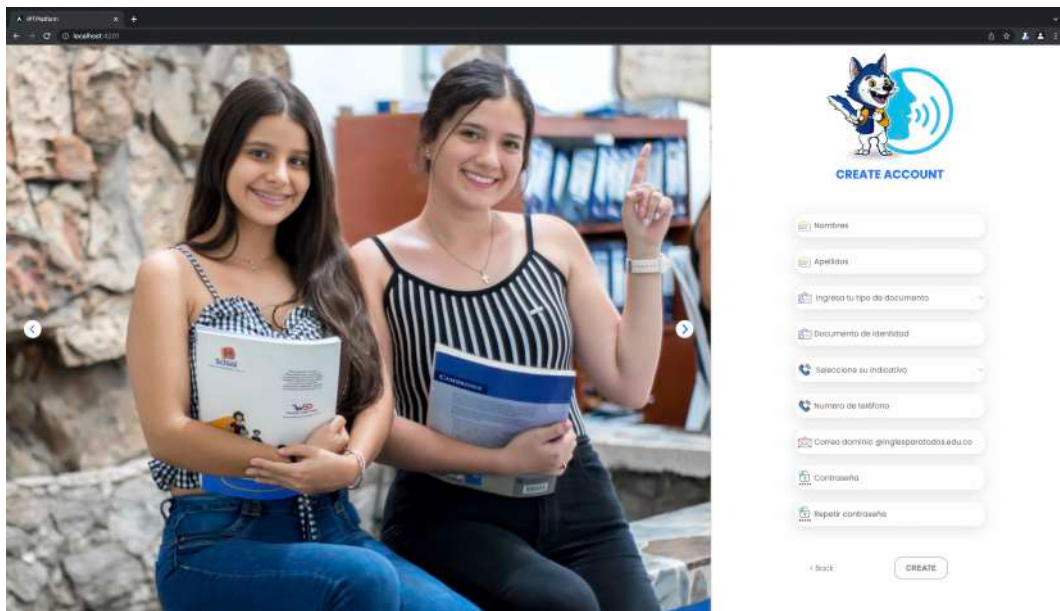


Figura 4.19: Vista de Register - IPT Platform.

Fuente: Autor

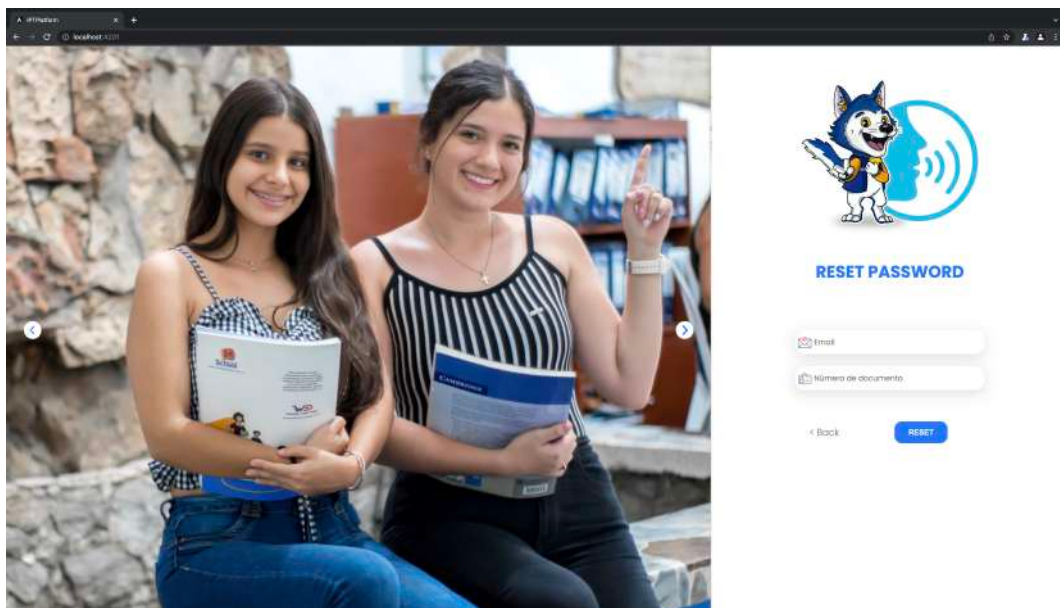


Figura 4.20: Vista de Forgot-Password - IPT Platform.

Fuente: Autor

4.3.3 Dashboard

En esta sección se encuentran varias vistas, sobre todo para el departamento académico, el cual es el más importante de todos y la prioridad del primer release.

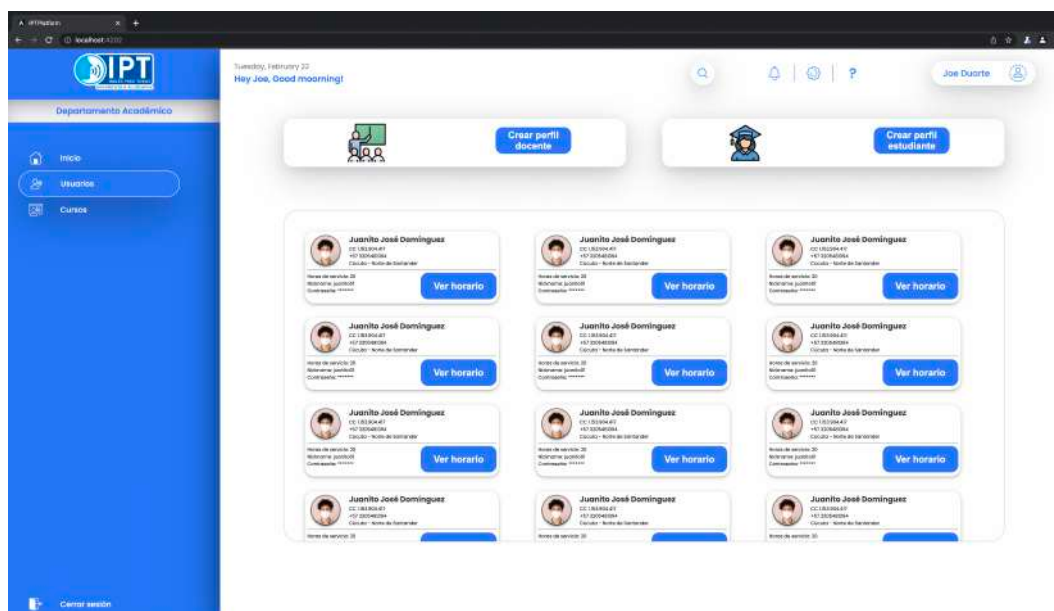


Figura 4.21: Vista de usuarios, departamento académico - IPT Platform.

Fuente: Autor

The screenshot shows the 'Crear teacher' form in the IPT Platform. The form is a modal window with the following fields:

- Nombre (Name)
- Apellido (Surname)
- Correo electrónico (Email)
- Selecciona el rol (Select role)
- Selecciona su institución (Select institution)
- Ingresa tu tipo de documento (Enter your document type)
- Documento de identidad (Identity document)
- Selecciona su departamento (Select department)
- Ingresa su ciudad de residencia (Enter your residence city)
- Fecha de servicio (Service date)
- Contraseña (Password)
- Repetir contraseña (Repeat password)

A 'Crear' button is located at the bottom of the form. The background shows a dashboard with a sidebar and a list of teachers.

Figura 4.22: Formulario de creación de usuarios - IPT Platform.

Fuente: Autor

The screenshot shows the 'Cursos activos' view in the IPT Platform. It displays a table of active courses with the following columns:

- TEACHER
- HORARIO
- PLAN
- TIPO
- CONVENIO
- SALÓN
- FECHA DE INICIO
- FECHA DE FIN

The table is filtered by 'Cursos activos' and 'B1' level. The data is as follows:

TEACHER	HORARIO	PLAN	TIPO	CONVENIO	SALÓN	FECHA DE INICIO	FECHA DE FIN
Oscar Donado	2:00PM A 4:00PM	Academy	Intensivo	Universidad	New York	04/06/2021	04/08/2021
Vladimir Velarde	2:00PM A 4:00PM	Social Kids	PT Kids (Semestral)	Sin convenio	Principito	04/06/2021	04/08/2021
Eber Rayona	2:00PM A 4:00PM	Universities	Semestral	Universidad de Pamplona	Washington	04/06/2021	04/08/2021
Eber Rayona	2:00PM A 4:00PM	Business	Técnica B1	Sin convenio	London	04/06/2021	04/08/2021
Oscar Donado	2:00PM A 4:00PM	Academy	Intensivo	Universidad	New York	04/06/2021	04/08/2021
Vladimir Velarde	2:00PM A 4:00PM	Social Kids	PT Kids (Semestral)	Sin convenio	Principito	04/06/2021	04/08/2021
Eber Rayona	2:00PM A 4:00PM	Universities	Semestral	Universidad de Pamplona	Washington	04/06/2021	04/08/2021
Eber Rayona	2:00PM A 4:00PM	Business	Técnica B1	Sin convenio	London	04/06/2021	04/08/2021
Oscar Donado	2:00PM A 4:00PM	Academy	Intensivo	Universidad	New York	04/06/2021	04/08/2021
Vladimir Velarde	2:00PM A 4:00PM	Social Kids	PT Kids (Semestral)	Sin convenio	Principito	04/06/2021	04/08/2021
Eber Rayona	2:00PM A 4:00PM	Universities	Semestral	Universidad de Pamplona	Washington	04/06/2021	04/08/2021
Eber Rayona	2:00PM A 4:00PM	Business	Técnica B1	Sin convenio	London	04/06/2021	04/08/2021

Figura 4.23: Vista de cursos, departamento académico - IPT Platform.

Fuente: Autor

Figura 4.24: Formulario de creación de cursos - IPT Platform.

Fuente: Autor

Figura 4.25: Formulario de edición de cursos - IPT Platform.

Fuente: Autor

5 Conclusiones

El objetivo primordial de esta tesis ha sido desarrollar metodologías ágiles para la optimización de las plataformas usadas por la academia Inglés Para Todos, obteniendo gratificantes resultados por parte de los propietarios de todos los productos requeridos, los resultados esperados por el aprendiz han superado las expectativas, de modo que se da el visto bueno por parte de los productores para que pueda continuar participando con un buen rol en los mismos proyectos y los que se requieran a medida que crecen las plataformas. Con la realización de todos los proyectos se demuestra que es posible aplicar tecnologías eficientes al entorno educativo para brindar un mejor servicio, no solo para los estudiantes si no también para los empleados y miembros de la empresa, generar un ambiente tecnológico y eficaz ha sido uno de los objetivos principales, llevando a cabo cada uno de los proyectos TI requeridos.

- Ha sido demostrada la importancia de la ruta de aprendizaje como primer objetivo, ejecutar un recorrido de aprendizaje basado en cursos cortos como secciones de capacitación general genera una preparación importante para el desarrollador, así mismo conocer la lógica de negocio de la empresa y el entorno tecnológico que se llevaba hasta ese momento da mucha más claridad al momento previo de formar parte del equipo de desarrollo de manera técnica. La capacitación constante conllevó al buen desarrollo de las plataformas requeridas y a satisfactorias entregas de las mismas.
- El desarrollo de la plataforma para pruebas EKT resultó ser un gran producto, ya que ha podido impactar en el servicio de pruebas para los estudiantes de la universidad Francisco de Paula Santander. Las innovaciones requeridas fueron entregadas y aprobadas por los propietarios del mismo producto, dando sin dudas un visto bueno a las implementaciones de API de Jitsi para video conferencias, visualización de pruebas en tiempo real, verificación de listados de grupos, y muchas más actualizaciones que fueron requeridas previamente.
- IPT World nace desde la necesidad de contar con una aplicación propia de la empresa para los estudiantes de la misma, el uso de Flutter para el primer desarrollo de este proyecto ha sido muy profesional y bien recibido como primera entrega, a pesar de no estar en producción, se aplicaron buenas metodologías para la arquitectura de software del aplicativo, usando metodologías de atomic design y arquitectura BLoC la cual separa la lógica de negocio de la interfaz gráfica.

- Se ha hecho una entrega parcial de la plataforma principal de la empresa, aplicando dentro del desarrollo todos los conocimientos adquiridos a lo largo de la ruta de aprendizaje y de la experiencia de previos proyectos, buenas prácticas de escritura de código y una gestión de calidad bien definida y practicada; se ha manejado muy bien el servidor adquirido y la base de datos principal tiene buena conexión con la arquitectura general de todo el sitio web.
-

6 Bibliografía

[1] Systems Groups. Ingeniería de software: “La ingeniería de software ¿qué es y qué utilidad tiene?”. En línea. 2019. Disponible en: (<https://systemsgroup.es/tecnologiasde-la-informacion/la-ingenieria-de-software-que-es-y-que-utilidad-tiene/32363/>)

[2] Voigtmann Informations Technologien. “Desarrollo de software”. En línea. 2005. Disponible en: (<https://www.voigtmann.de/es/desarrollo-de-software/>).

[3] Escuela IT. Materias. “Desarrollo web disciplinas enfocadas en el desarrollo de sitios y aplicaciones web”. En línea. 2021. Disponible en: (<https://escuela.it/materias/desarrollo-web>).

[4] TechTarget. Desarrollo de aplicaciones móviles. “Qué es el desarrollo de aplicaciones móviles”. En línea. agosto 2021. Disponible en: (<https://searchdatacenter.techtarget.com/es/definicion-de-aplicacionesmoviles>).

[5] Platzi. Blog. “Bases de datos: qué son, qué tipos existen y cómo funcionan”. En línea. 2017. Disponible en: (<https://platzi.com/blog/bases-de-datos-que-son-que-tiposexisten/>).

[6] Tic Portal. Glosario tic. “Servidores”. En línea. 08/07/2019. Disponible en: (<https://www.ticportal.com/tic/servidores>).

[7] On Desarrollo. “¿QUÉ ES UN FRAMEWORK Y PARA QUÉ SE UTILIZA?”. En línea. 2021. Disponible en: (<https://ondesarrollo.com/que-es-un-framework-y-para-que-seutiliza/>).

[8] Open Webinars. Blog de programación y sistemas. Desarrollo web. “Qué es la programación orientada a objetos”. En línea. 11 de junio de 2019. Disponible en: (<https://openwebinars.net/blog/que-es-la-programacion-orientada-objetos/>).

[9] Atlassian. Bit Bucket. Tutorials. “¿Qué es Git?”. En línea. 2022. Disponible en: (<https://www.atlassian.com/es/git/tutorials/what-is-git>).

[10] Xataka. Xataka Basics. “Qué es Github y qué es lo que ofrece a los desarrolladores”. En línea. 30 de octubre de 2019. Disponible en: (<https://www.xataka.com/basics/que-github-que-le-ofrece-a-desarrolladores>).

[11] Platzi. Curso práctico de SQL. “Breve historia de SQL”. En línea. 2020. Disponible en: (<https://platzi.com/clases/2059-practico-sql/32987-breve-historia-de-sql/>).

[12] Styde. Eloquent ORM. “Aprende a usar Eloquent el ORM de Laravel”. En línea. 15 de mayo de 2015. Disponible en: (<https://styde.net/aprende-a-usar-eloquent-el-orm-de-laravel/>).

[13] Platzi. Cursos. Curso de Postman. “Postman es la plataforma colaborativa para el desarrollo y testing de APIs. Construye y gestiona múltiples APIs tuyas o de terceros”. En línea. 2020. Disponible en: (<https://platzi.com/cursos/postman/>).

[14] Flutter. Flutter dev. “Build apps for any screen”. En línea. 2022. Disponible en: (<https://flutter.dev/>)

[15] Scrum México. Informato. “Escribiendo historias de usuario”. En línea. 2 de agosto de 2018. Disponible en: (<https://scrum.mx/informato/historias-de-usuario>).

[16] Vera Balderas, S., Moreno Tapia, J. (2022). Las competencias docentes en la enseñanza del idioma inglés en un contexto universitario durante la pandemia de la COVID-19. *Revista Universidad y Sociedad*, 14(S1), 151-160.

[17] L. E. Cabezas Arévalo, «Nivel de satisfacción en el aprendizaje del idioma inglés en un entorno totalmente digital por tiempos de Covid-19», *AP*, vol. 3, n.º 3.1, pp. 116–128, ago. 2021.
