



SISTEMA DE BEDELÍA

PROYECTO TECNÓLOGO INFORMÁTICO

AUTORES

Álvarez, Juan

Garrido, Lucas

Morales, Sebastián

TUTOR

Escobar, Nicolás

Índice

Índice.....	1
Agradecimientos.....	3
Resumen.....	4
Palabras clave.....	4
1. Introducción	5
2. Objetivos planteados y resultados esperados	6
3. Marco teórico	7
3.1. Web.....	7
3.2. HTML	8
3.3. CSS.....	8
3.4. JavaScript	8
3.5. Framework	8
3.6. TypeScript.....	8
3.7. PHP.....	9
3.8. Angular.....	9
3.9. RESTful.....	9
3.10. PWA (Progressive Web App)	9
3.11. Arquitecturas	10
3.11.1. Modelo cliente - servidor	10
3.11.2. Arquitectura en capas	10
3.12. Mobile.....	11
3.12.1. Aplicaciones nativas.....	11
3.12.2. Aplicaciones web	12
3.12.3. Aplicaciones web nativas	12
3.13. Plataformas Mobile.....	12
3.13.1. Android	12
3.13.2. IOS	13
4. Análisis del problema.....	15
4.1. Requerimientos del sistema	15
4.2. Vista del modelo de casos de uso	16
4.2.1. Casos de uso relevantes a la arquitectura	17
4.2.1.1. Inscripción de postulante a carrera	17
4.2.1.2. Creación de carrera.....	17
4.2.1.3. Ver el grafo de previaturas.....	17
4.2.1.4. Asignar docente a curso o examen.....	17

4.2.1.5.	Control de asistencia	17
4.2.1.6.	Inscripción a curso.....	18
4.3.	Dominio del problema.....	19
4.4.	Arquitectura del sistema	20
4.4.1.	Trazabilidad desde el modelo de casos de uso al modelo de diseño.....	20
4.5.	Vista del modelo de diseño.....	20
4.5.1.	Descomposición en subsistemas.....	21
4.5.2.	Web SPA.....	21
4.5.3.	Progressive Web App	21
4.5.4.	API REST	21
4.5.5.	Persistencia.....	21
4.6.	Trazabilidad desde el modelo de diseño al modelo de implementación.....	22
4.6.1.	Vista del modelo de distribución	22
5.	Implementación	23
5.1.	Metodologías de trabajo	23
5.2.	Entorno de desarrollo	24
5.3.	Entorno de ejecución.....	25
5.4.	Tecnologías aplicadas.....	25
5.4.1.	Capa de persistencia	25
5.4.2.	Capa de negocio	26
5.4.3.	Capa de presentación.....	26
5.4.4.	Web.....	26
5.4.5.	Aplicación PWA	27
5.5.	Problemas encontrados.....	27
5.6.	Horas dedicadas	28
5.7.	El sistema desarrollado	29
5.7.1.	Aplicación web.....	29
5.7.2.	Aplicación móvil.....	37
6.	Conclusiones	38
7.	Trabajo a futuro	39
8.	Referencias.....	40

Agradecimientos

Queremos agradecer a nuestras familias, amigos y compañeros de clase, quienes estuvieron presentes durante la realización de este proyecto, brindándonos apoyo, aconsejándonos y sabiendo entender las largas jornadas delante de la computadora.

También queremos agradecer a nuestro tutor Nicolás Escobar, quien a lo largo de estos meses nos guió, dándonos consejos, respondiendo nuestras dudas y apoyándonos en nuestra toma de decisiones, lo cual fue clave para culminar el proyecto en tiempo y forma.

Resumen

En el marco de la asignatura Proyecto se presentará a continuación un resumen para mayor entendimiento del mismo y a lo largo del documento se darán más detalles al respecto.

UTEC es la Universidad Tecnológica del Uruguay, una propuesta de educación terciaria universitaria pública de perfil tecnológico, orientada a la investigación y la innovación. Comprometida con los lineamientos estratégicos del país, abierta a las necesidades del Uruguay productivo, que tiene entre sus objetivos centrales hacer más equitativo el acceso a la oferta educativa especialmente en el interior del país. Uno de sus rasgos más novedosos es su enfoque en la construcción colectiva de conocimientos, con altos estándares de calidad de gestión y excelencia académica.

Se plantea la realización de un sistema de bedelía para dicha institución, el cual permitirá gestionar tanto la información de la propia universidad, así como la de los estudiantes de manera fácil y eficiente.

El objetivo planteado es diseñar e implementar el sistema poniendo en práctica los conocimientos adquiridos y haciendo uso de las tecnologías aprendidas durante la carrera. Para ello, se ha elegido utilizar las tecnologías de PHP, Angular, MySQL y Docker.

El sistema a desarrollar consta de un sitio web desde el cual los administradores del sistema, el personal administrativo, los docentes y los estudiantes podrán realizar diversas tareas. También se desarrollará una aplicación móvil basada en tecnologías web, en la cual los estudiantes podrán gestionar su actividad.

Palabras clave

MySQL, PHP, Angular, PWA, Docker

1. Introducción

La Universidad Tecnológica del Uruguay (UTEC) es una institución uruguaya de educación terciaria, universitaria y pública, de perfil tecnológico y orientada a la investigación e innovación.

La UTEC comenzó a funcionar a principios de 2013. Actualmente imparte once carreras en siete departamentos de Uruguay: Colonia, Soriano, Paysandú, Río Negro, Durazno, Rivera y San José, y en el año 2020 se superaron los 3.000 estudiantes.

Creada por la Ley 19.043 (28 de diciembre de 2012) como una universidad pública autónoma, la UTEC está integrada por Institutos Tecnológicos Regionales (ITR), y su sede central se encuentra ubicada en el interior del país. Cada ITR está especializado en una determinada área productiva, sin perjuicio de poder brindar otros cursos terciarios y terciarios universitarios.

El problema planteado y los objetivos propuestos, llevan a la necesidad de entender en mayor profundidad la realidad de la institución educativa, de su gestión, la administración de sus sedes y las necesidades de sus funcionarios y estudiantes. Por tanto es necesario brindar una solución informática a los problemas planteados.

El siguiente capítulo, titulado Objetivos Planteados y Resultados Esperados, hace referencia a las necesidades planteadas por el cliente y los resultados que se esperan al final del proyecto. En el capítulo Marco teórico, se presentan los conceptos que servirán de base teórica para comprender los capítulos posteriores. En el siguiente capítulo, Análisis del Problema, se detalla la solución referente a las aplicaciones planteadas. Luego en el capítulo Implementación, se detalla el proceso de desarrollo, los problemas encontrados y la evolución del proyecto. A continuación en el capítulo Conclusiones se analizan los resultados obtenidos, el trabajo realizado y, finalmente, en el capítulo Trabajo a futuro se plantean eventuales mejoras que se pueden implementar e incorporar al sistema.

2. Objetivos planteados y resultados esperados

El objetivo del proyecto consiste en analizar y desarrollar un sistema que permita gestionar la información de la institución y oficie de bedelía virtual para sus estudiantes. Como parte del análisis se espera estudiar las tecnologías PHP y Angular para el desarrollo de la plataforma web, la tecnología PWA para el desarrollo de una aplicación para dispositivos móviles y Docker para el despliegue del sistema construido.

Como caso de estudio, se nos plantea desarrollar un sistema basado en la web que será utilizado tanto por funcionarios de la institución como por los estudiantes para realizar diversas gestiones, es por este motivo que el sistema debe ser amigable, atractivo, confiable y seguro.

Se espera la realización de una aplicación móvil que esté disponible tanto en Android como en IOS y que brinde a los estudiantes de la institución la posibilidad de gestionar su escolaridad y sus inscripciones.

Como solución final se espera un sistema funcionando respondiendo al problema planteado.

3. Marco teórico

Esta sección formaliza e intenta expresar de forma clara las tecnologías mencionadas en los siguientes capítulos.

Su propósito es presentar un marco conceptual que servirá de base teórica para comprender el problema desde el punto de vista técnico.

3.1. Web

El término Web¹ se utiliza para denominar a una red informática, y particularmente a Internet. A partir de dicho concepto se pueden encontrar términos como página web, servidor web, navegador web, entre otros.

La Internet, también conocida como World Wide Web (WWW), se puede entender como una red mundial de computadoras que permite el acceso a diferentes tipos de información, así como también, el intercambio de diferentes tipos de archivos a través de sitios y páginas web, publicadas por servidores web, accedidas por navegadores web.

Todos los conceptos descritos anteriormente, son abarcados por lo que se denomina Tecnologías Web, siendo estas las herramientas que facilitan el desarrollo de sitios web. Estas permiten lograr mejores y más rápidos resultados, principalmente útiles a la hora de escalar las soluciones alcanzadas.

En la actualidad, el acceso a contenidos mediante sitios web se puede realizar desde múltiples y variadas plataformas, dejando de lado la antigua necesidad de contar con un ordenador personal para dichos fines. Los teléfonos inteligentes o smartphones, tabletas y otros dispositivos encontrados en el hogar y en la vida cotidiana son algunos ejemplos al respecto. Esto se debe al crecimiento exponencial que ha tenido la Internet, que es directamente proporcional al crecimiento de la necesidad de acceder a los contenidos que ésta brinda. Por esta razón ha dejado de ser una simple herramienta de consulta para convertirse en parte de la vida cotidiana y conllevando a la necesidad de acceder a información o aplicaciones desde cualquier lugar y en cualquier momento.

3.2. HTML

HTML², siglas en inglés de HyperText Markup Language ('lenguaje de marcas de hipertexto'), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.

3.3. CSS

CSS³ (siglas en inglés de Cascading Style Sheets), en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.

3.4. JavaScript

JavaScript⁴ (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

3.5. Framework

Un framework⁵, o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

3.6. TypeScript

TypeScript⁶ es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases.

3.7. PHP

PHP⁷ es un lenguaje de scripting de código abierto, del lado del servidor, con programación HTML integrada que se utiliza para crear páginas web dinámicas. Las ventajas de PHP son su flexibilidad y su alta compatibilidad con otras bases de datos. Además, PHP es considerado como un lenguaje fácil de aprender.

3.8. Angular

Angular⁸ es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

3.9. RESTful

Un servicio RESTful⁹ hace referencia a un servicio web que implementa la arquitectura REST (REpresentational State Transfer), la cual podríamos decir que es un conjunto de principios, o maneras de hacer las cosas, que define la interacción entre distintos componentes, es decir, las reglas que dichos componentes tienen que seguir. En dicha arquitectura, un servidor provee acceso a recursos y los clientes REST acceden y presentan los recursos.

Estos servicios son livianos, escalables y fáciles de mantener, por ello son considerados el estilo de arquitectura más adecuado para el desarrollo de aplicaciones web. También existen otros servicios, tales como SOAP (Simple Object Access Protocol) que están basados en lenguaje XML, permiten la interacción entre varios dispositivos y tienen la capacidad de transmitir información compleja.

3.10. PWA (Progressive Web App)

Una aplicación web progresiva (PWA¹⁰) es un tipo de software de aplicación que se entrega a través de la web, creada utilizando tecnologías web comunes como HTML, CSS y JavaScript. Está destinado a funcionar en cualquier plataforma que use un navegador compatible con los estándares.

3.11. Arquitecturas

El concepto de arquitectura de software¹¹ representa estructuralmente un diseño de alto nivel del sistema y tiene como objetivo ayudar a satisfacer los atributos de calidad y a su vez servir como guía para el desarrollo. Es por este motivo que idealmente se crea en etapas tempranas del desarrollo. A continuación, se presentan los modelos de arquitectura más conocidos.

3.11.1. Modelo cliente - servidor

En las aplicaciones web que utilizan el modelo cliente-servidor¹², generalmente el usuario final, o en este caso cliente, inicia una comunicación cuando desea acceder o interactuar con el servicio. El cliente en este caso entra en contacto con el servidor, quien le brinda la información solicitada, o eventualmente reacciona ante la petición del cliente. En este modelo se le llama cliente al programa o dispositivo que entra en contacto con el servidor remoto a efectos de lograr la interacción deseada. No se debe confundir el cliente con el usuario que utiliza el sistema.

3.11.2. Arquitectura en capas

La arquitectura en capas¹³ propone separar la lógica del negocio del diseño de la interfaz de usuario. De esta forma se facilita el mantenimiento del sistema, y el desarrollo inicial del mismo, permitiendo abstraerse de la solución y concentrarse en una capa puntual.

Generalmente se habla de tres capas, éstas son:

- Capa de presentación:

Es la cara visible de la aplicación, la que permite al usuario hacerse de la información solicitada, o interactuar con el servicio ofrecido. En ella el usuario ingresa datos que posteriormente son enviados a la siguiente capa, a la vez que recibe los resultados obtenidos de la misma, la capa de negocio.

- Capa de negocio:

Cada negocio o escenario plantea y define sus propias reglas, a las cuales se les denomina reglas de negocio. Estas reglas son el núcleo de

la aplicación o sistema y son las que definen las funcionalidades que se ofrecen en el mismo. De ahí el nombre de esta capa. Para sustentar estas reglas, es aquí donde el sistema ejecuta, procesa y resuelve las peticiones de cada cliente, devolviendo datos en caso de ser necesario. Para poder realizar estas acciones, es necesario interactuar con la siguiente capa de la arquitectura, la capa de datos.

- Capa de datos:

La capa de acceso a datos, o simplemente capa de datos, es la encargada de procesar las peticiones de datos, y devolverlas a la capa anterior, así como actualizar o agregar información al juego de datos. Por este motivo es la encargada de comunicarse con los datos mismos, los que se suelen persistir, ya sea en sistemas de archivos, base de datos, o algún otro modo de almacenamiento.

3.12. Mobile

Hoy en día, el término Mobile¹⁴ no solo aplica a teléfonos móviles en sí mismos, sino que también abarca otros dispositivos, como Tablets, Smartphones, Smart TV entre otros, que permiten acceder a aplicaciones en cualquier momento y desde casi cualquier ubicación. Este avance tecnológico ha conllevado a un incremento exponencial de las aplicaciones que soportan estos dispositivos, por lo que el mercado de dichas aplicaciones móviles o Apps no ha parado de crecer.

Se pueden encontrar 3 tipos de aplicaciones móviles, las cuales se pueden adaptar mejor unas que otras según las circunstancias. Estas se describen a continuación.

3.12.1. Aplicaciones nativas

Este tipo de aplicación se desarrolla sobre Software Development Kit o SDK, que es un conjunto de herramientas que ayudan a programar la aplicación para un entorno tecnológico específico. Al existir distintas plataformas, como Android o iOS y cada una de ellas cuenta con un sistema totalmente diferente, si se desea tener un sistema en todas las plataformas, es necesario crear varias Apps con el lenguaje del sistema operativo correspondiente.

Cuando se habla de desarrollo Mobile, habitualmente se hace referencia a aplicaciones nativas, ya que tienen como ventaja, respecto a otro tipo de aplicaciones móviles, la posibilidad de utilizar funciones del hardware del dispositivo como, por ejemplo, GPS, cámara, almacenamiento, etc.

3.12.2. Aplicaciones web

Una aplicación web o web App es aquella que se ejecuta dentro del propio navegador web del dispositivo a través de una URL. Tiene la ventaja de no requerir instalación en el dispositivo, lo que, a su vez, facilita a nivel de desarrollo, ya que permite programar una única App independientemente del sistema operativo en el que se vaya a ejecutar.

3.12.3. Aplicaciones web nativas

Las aplicaciones web nativas, también conocidas como aplicaciones híbridas, son aquellas que comparten características de las aplicaciones web y las aplicaciones nativas. Se podría decir que es un tipo de App que recoge lo mejor de los dos tipos de aplicaciones mencionados anteriormente, ya que se desarrollan para ser ejecutadas a través de un navegador al igual que la web App, pero a su vez permiten cierta integración con el dispositivo, permitiendo acceder a algunas de sus características de hardware.

3.13. Plataformas Mobile

3.13.1. Android

Android¹⁵ es un sistema operativo desarrollado, desde la ideología OpenSource, por la compañía Google. Está basado en GNU/Linux y enfocado a dispositivos móviles de todo tipo, ya sean teléfonos móviles, tabletas e incluso mini ordenadores portátiles.

Ventajas:

- Android es un sistema operativo de código abierto y es apto para cualquier dispositivo. Esto permite que distintas compañías lo utilicen como base, aplicando sus propias capas de personalización.
- Es fácil encontrar firmwares “no oficiales” con actualizaciones más recientes, las cuales son bastante simples de aplicar a los dispositivos.

- Es bastante más sencillo, respecto a otras plataformas, el acceso a instalar aplicaciones de terceros “no homologadas” por Google. Esto permite probar aplicaciones de todo tipo, lo que para un desarrollador es una ventaja.

Desventajas:

- Una de las principales desventajas es que, al poder aplicar capas de personalizaciones variadas por las distintas compañías, muchas veces las actualizaciones no solo no llegan en tiempo y forma, si no que algunas veces son discontinuadas para dar soporte a terminales más “nuevos”.
- Otra desventaja es que al tener fácil acceso para instalar aplicaciones “no homologadas” se abre una puerta de acceso a la información del dispositivo para desarrolladores mal intencionados.

3.13.2. IOS

iOS¹⁶ es un sistema operativo propiedad de Apple y está orientado específicamente para su uso mediante dispositivos móviles de dicha compañía, y es una variante del Mac OS X, que es el sistema operativo para computadoras de Apple. iOS, al igual que OS X, está basado en Unix, por lo tanto, guarda muchas similitudes con Linux (sistema operativo basado en el mismo sistema), pero a diferencia de este último, iOS y OS X son sistemas operativos propietarios de Apple y están cerrados para el uso específico de sus dispositivos, mientras que Linux es de código abierto y puede ser utilizado en una infinidad de dispositivos.

Ventajas:

- Un buen punto a favor de iOS es que aun siendo un sistema operativo puro y exclusivamente para dispositivos de la marca Apple, ésta libera su kit de desarrollo para que pueda ser mejorado por otros desarrolladores, haciendo que, de esta forma, la App Store sea considerada la tienda con mayor cantidad de aplicaciones disponibles.
- El tiempo de respuesta de las aplicaciones es muy bueno.
- Interfaz intuitiva.

Desventajas:

- La principal desventaja es que, al ser propiedad exclusiva de Apple, solo se puede disponer adquiriendo un dispositivo de la mencionada compañía.
- Otra desventaja es a nivel del uso de su multitarea, ya que reserva la misma para algunos procesos fundamentales y no para cualquier tipo de aplicación.
- Para utilizar el kit de desarrollo y subir aplicaciones a su tienda, es necesario contar con una Mac

4. Análisis del problema

Se nos solicita diseñar e implementar un sistema que permita la gestión de una institución educativa pública y de nivel nacional. Por este motivo se crea una plataforma web, desde la cual los diferentes usuarios podrán realizar acciones en base a su rol en el sistema. Entre otras acciones, los administradores del sistema podrán registrar las diferentes sedes, cursos y carreras, en cada sede los funcionarios administrativos tendrán la posibilidad de revisar actas e inscripciones a carreras, los docentes dispondrán de la información de los estudiantes para realizar el control de asistencia y el ingreso de calificaciones, mientras que los estudiantes serán capaces de consultar su escolaridad e inscribirse a cursos y exámenes.

También se construye una aplicación de tipo PWA que permite a los estudiantes consultar su escolaridad e inscribirse a cursos y exámenes.

4.1. Requerimientos del sistema

Esta sección tiene el propósito de oficiar como medio de comunicación entre el cliente y los desarrolladores, sirviendo como base contractual en la que se especificarán los requerimientos funcionales y no funcionales a implementar.

El objetivo de este proyecto, es permitir gestionar y administrar los diferentes aspectos que componen la administración de la institución educativa UTEC.

Se implementará una aplicación web, en la que se permitirá a los estudiantes y funcionarios realizar acciones dependiendo de su rol, a través de una interfaz amigable y de fácil uso. Los estudiantes podrán gestionar sus inscripciones a cursos y exámenes, mientras que los funcionarios podrán realizar tareas administrativas sobre el sistema en forma rápida, sencilla y efectiva.

Así mismo se implementará también una aplicación móvil exclusiva para estudiantes de la institución, la cual permitirá realizar inscripciones a cursos y gestión de escolaridad.

4.2. Vista del modelo de casos de uso

A continuación, se muestra el diagrama de todos los casos de uso, que representa la relación entre los actores y las funcionalidades del sistema. Luego se brindan detalles sobre los casos de uso relevantes a la arquitectura.

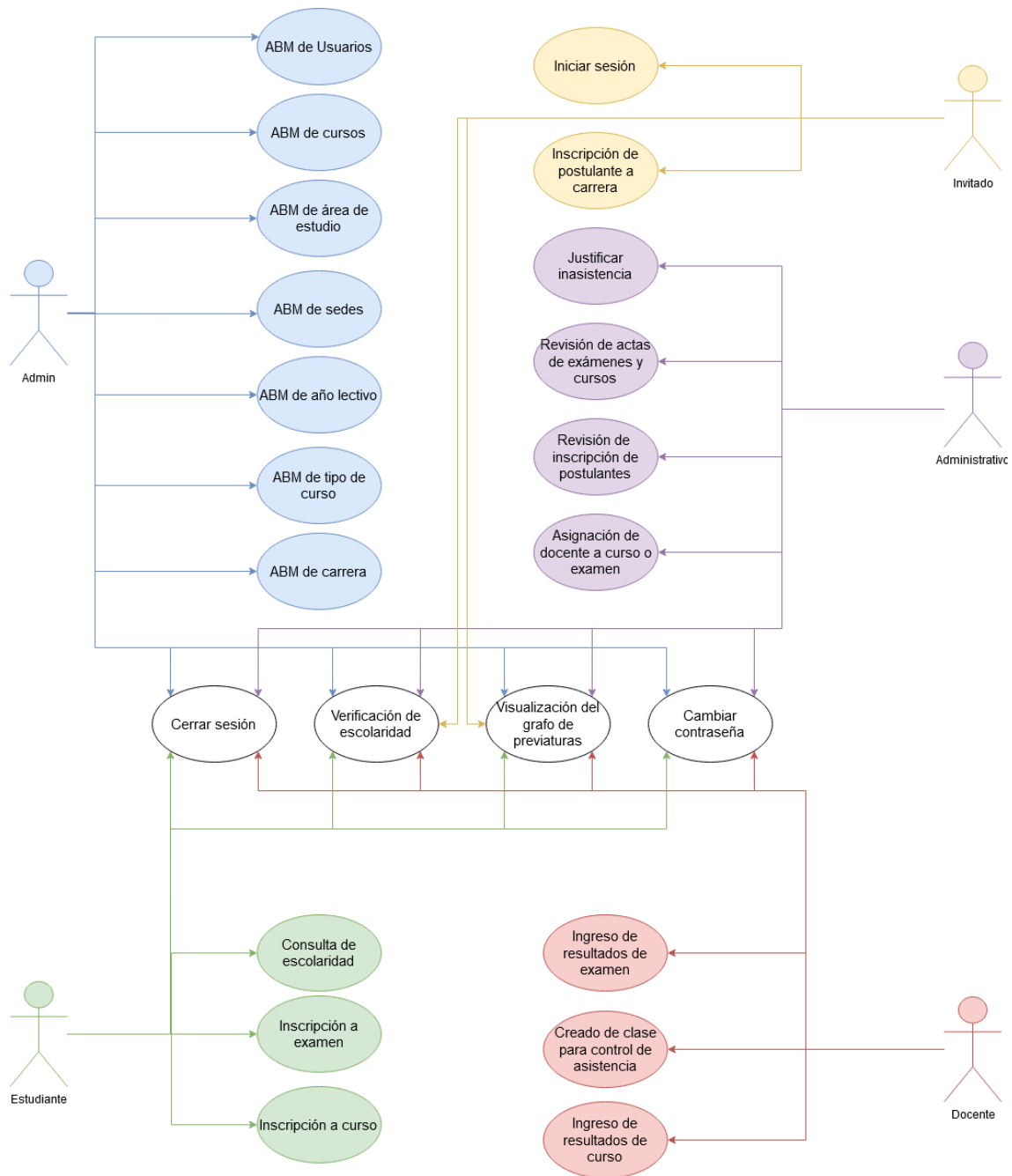


Imagen 1. Diagrama de Casos de Uso

4.2.1. Casos de uso relevantes a la arquitectura

Los casos de uso que se expondrán en esta sección se consideran relevantes dado que afectan a la mayor cantidad de capas de la aplicación. Estos casos fueron los primeros en ser modelados e implementados.

4.2.1.1. Inscripción de postulante a carrera

Permite a los usuarios (registrados o no) postularse para inscribirse a una carrera. La persona que se postula deberá completar un formulario con sus datos básicos y adjuntar la documentación solicitada.

4.2.1.2. Creación de carrera

Permite a los administradores del sistema registrar o modificar las carreras dictadas por la institución, estableciendo entre otros datos, los cursos que la componen, sus previaturas y en qué sedes se dicta.

4.2.1.3. Ver el grafo de previaturas

Permite que cualquier usuario vea información general de una carrera, las sedes en las que se dicta, los cursos que la conforman y el grafo de previaturas entre los mismos.

4.2.1.4. Asignar docente a curso o examen

Permite a los funcionarios administrativos de cada sede, asignar un docente ya registrado en el sistema a uno o más cursos o exámenes que se realicen en la sede en la que trabaja.

4.2.1.5. Control de asistencia

Permite a los docentes realizar el control de asistencia de los estudiantes inscriptos en los cursos que este dicta.

4.2.1.6. Inscripción a curso

Permite a los estudiantes inscribirse a los cursos correspondientes a la carrera que cursan. Esta acción se realizará desde la aplicación móvil.

4.4. Arquitectura del sistema

En esta sección se informa acerca de la arquitectura del sistema, donde se describe la misma utilizando el “Modelo de vistas de arquitectura 4+1”¹⁷. Este modelo describe cuatro vistas: vista lógica, vista de desarrollo, vista de proceso y vista física. Para poder hacer una trazabilidad a todos los componentes, se unifican las cuatro vistas mencionadas anteriormente a través de la vista de escenarios, la cual se corresponde con instancias de casos de uso.

4.4.1. Trazabilidad desde el modelo de casos de uso al modelo de diseño

La Vista Lógica se utiliza para marcar la trazabilidad entre el Modelo de Casos de Uso y el Modelo de Diseño, identificando los objetos y subsistemas de diseño que intervienen en el caso de uso y sus relaciones.

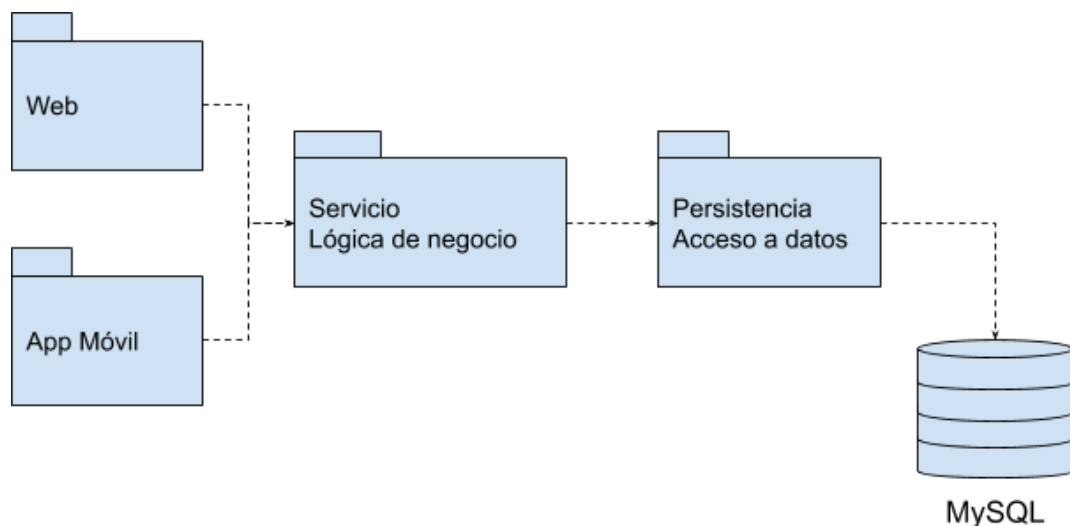


Imagen III. Diagrama de paquetes

4.5. Vista del modelo de diseño

El modelo de diseño de software es realmente un proceso de muchos pasos pero que se clasifican dentro de uno mismo. En general, la actividad del diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general del software, representaciones de interfaz y algoritmos.

4.5.1. Descomposición en subsistemas

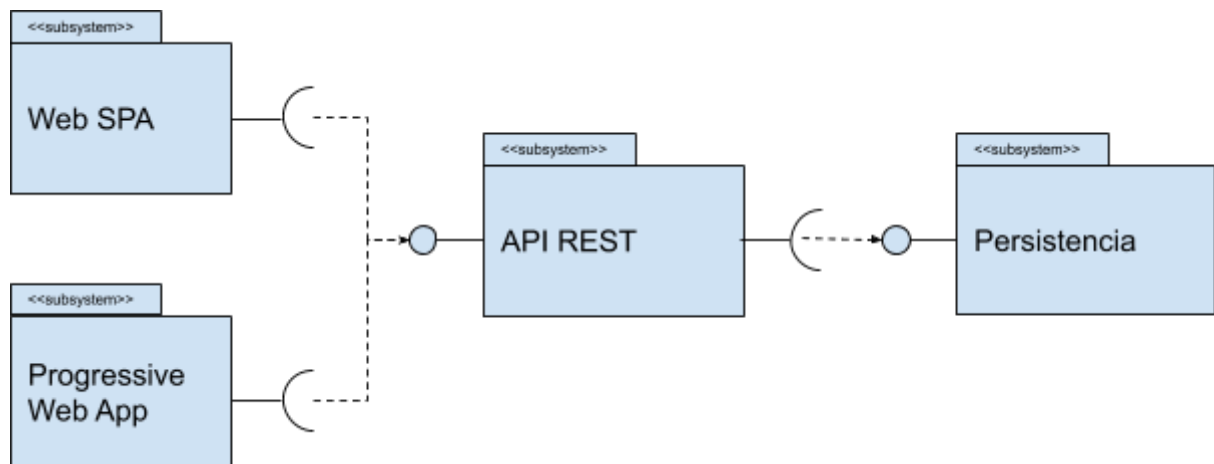


Imagen IV. Diagrama de Subsistemas

4.5.2. Web SPA

Este subsistema permite a los diferentes actores ejecutar las diferentes funcionalidades del sistema

4.5.3. Progressive Web App

Este subsistema consta de una PWA instalable en smartphones con sistema Android e IOS y estará destinada a los actores Estudiantes.

4.5.4. API REST

Este subsistema provee las funcionalidades del sistema como servicios, permitiendo a los subsistemas de presentación operar con estas funcionalidades.

4.5.5. Persistencia

Este subsistema se encarga del acceso a datos persistentes necesarios para que el sistema mantenga su información en el tiempo.

4.6. Trazabilidad desde el modelo de diseño al modelo de implementación

4.6.1. Vista del modelo de distribución

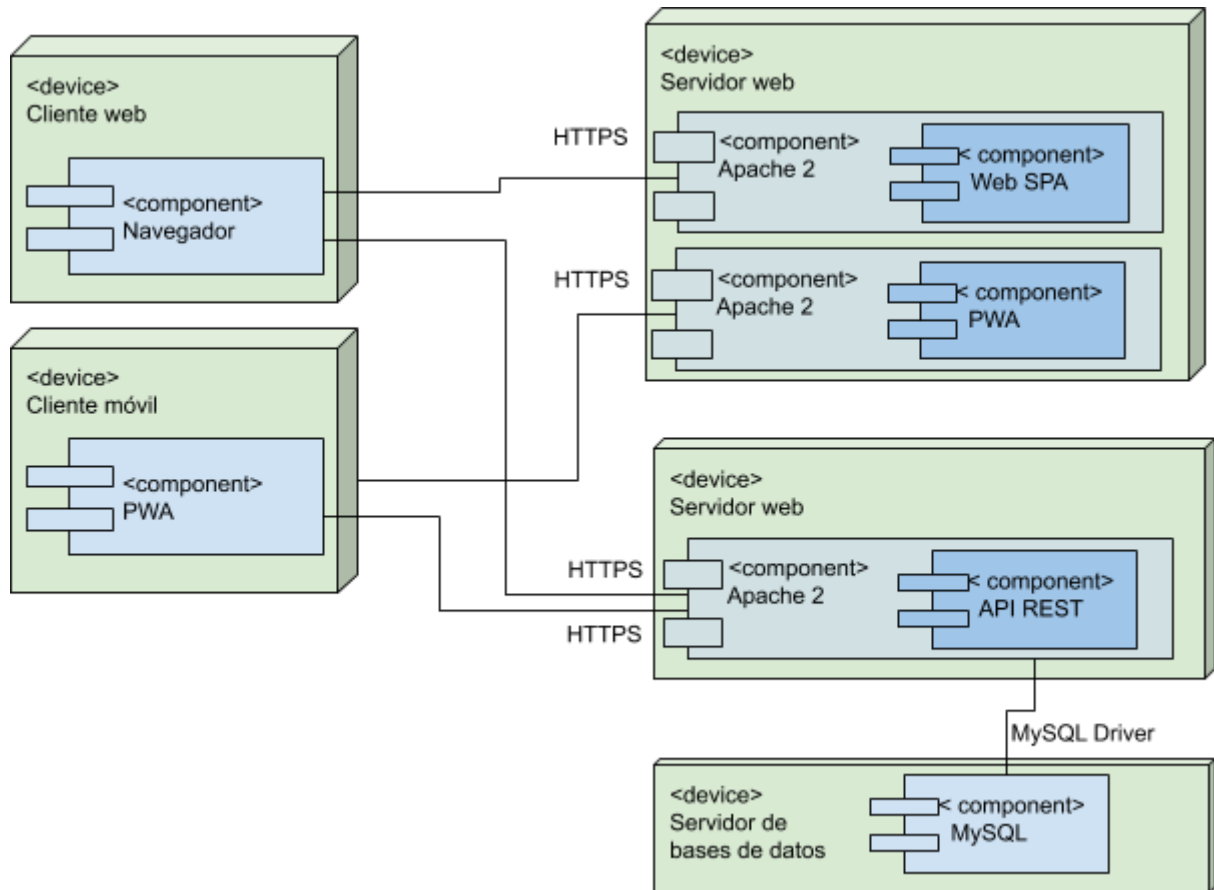


Imagen V. Modelo de Distribución

5. Implementación

En esta sección se brindan detalles sobre la metodología de trabajo utilizada, los entornos utilizados (desarrollo y ejecución), tecnologías aplicadas y problemas encontrados durante el transcurso del proyecto. También se detalla el proceso de implementación de las distintas funcionalidades de la aplicación, aplicando lo estipulado en las etapas de análisis y diseño.

5.1. Metodologías de trabajo

Para el desarrollo del proyecto, se decidió seguir la metodología iterativa incremental, definiendo cada semana como una iteración en la cual se deberá implementar determinados casos de uso. Los casos de uso elegidos para cada iteración, fueron definidos en base al tiempo estimado de implementación y la dependencia entre ellos. Para facilitar la coordinación entre los miembros del equipo acerca de las actividades a realizar en cada iteración, se utilizó la herramienta Trello¹⁸.

En lo que respecta a los integrantes del equipo, inicialmente se decidió que todos los integrantes trabajaran en modalidad full stack, implementando un caso de uso a la vez. Luego de un par de semanas y habiendo consultado con el tutor del proyecto, se definió que una mejor división de roles era que cada integrante trabajara solo en una capa, con lo cual, un integrante se dedicaría al desarrollo en Frontend¹⁹ y dos al desarrollo en Backend²⁰.

Los roles definidos fueron los siguientes:

- Project Manager:

Es el encargado de gestionar la planificación, el desarrollo del proyecto y vela por el cumplimiento de las tareas definidas. Es quien se encarga de documentar los avances y generar la documentación correspondiente recopilando la información brindada por el resto del equipo.

- Web Developer:

El encargado de estudiar y evaluar las tecnologías web vigentes para su posterior aplicación. También se encarga de diseñar las vistas utilizadas en la aplicación web.

- Backend Developer:

Es el encargado de estudiar las posibles tecnologías aplicables en las capas de persistencia y negocio de la aplicación, definir los protocolos adecuados para la seguridad, y generar los servicios que luego serían consumidos por la capa de presentación.

A pesar de que cada miembro del equipo tenía un rol asociado, en ocasiones eran necesarias consultas y la toma de decisiones grupales, independientemente de la capa de la que se tratase.

5.2. Entorno de desarrollo

El entorno de desarrollo utilizado, fue configurado sobre los sistemas operativos utilizados por los integrantes del equipo (Windows 10²¹ y Kubuntu 20.04²²). En ambos casos se documentó la configuración realizada dejando establecidas las instrucciones a seguir para replicar el entorno en caso de ser necesario.

La configuración mencionada incluye la instalación de:

- Visual Studio Code²³, el cual es un editor de código multilenguaje, que fue utilizado tanto para el desarrollo Backend como Frontend. A dicho editor, cada integrante le instaló plugins de su preferencia.
- PHP v7.4, el cual es el lenguaje de programación en el que se realizó la implementación del Backend.
- Composer²⁴, el cual es un manejador de dependencias para proyectos que utilizan el lenguaje de programación PHP.
- NodeJS²⁵, que permite la ejecución de código JavaScript fuera del navegador.
- NPM²⁶, que es un manejador de dependencias como Composer, pero para JavaScript
- Angular CLI²⁷, es la línea de comandos del framework de desarrollo web Angular, la cual permite generar el proyecto y ejecutar la aplicación durante su desarrollo.

- Docker²⁸, el cual es un software de contenerización, que en nuestro caso se utilizó durante el desarrollo para contener el servidor de bases de datos MySQL²⁹ y el gestor gráfico phpMyAdmin³⁰. Esto hizo que no fuera necesario instalar estos últimos en el sistema operativo.
- Git³¹, el cual permite el manejo de versiones del proyecto, y que en conjunto con GitHub³² permite compartir el código entre los desarrolladores.

5.3. Entorno de ejecución

Para la puesta en producción del sistema desarrollado, solo es necesario que el servidor cuente con Docker v19.03 y docker-compose v1.25 (v1.19 para Windows), ya que todos los componentes del sistema pueden ser contenerizados en contenedores Docker. A pesar de que Docker puede ser instalado en cualquier sistema operativo para servidor, es recomendable que se utilice sobre un sistema GNU/Linux para obtener un mejor rendimiento.

5.4. Tecnologías aplicadas

Esta sección brinda detalles de las tecnologías aplicadas para resolver las distintas funcionalidades de la aplicación. Se utilizó una implementación por capas, con una arquitectura cliente – servidor.

El objetivo primordial de este estilo es separar la capa de lógica de negocios y la capa de presentación. Su ventaja principal es que el desarrollo se puede llevar a cabo en varios niveles y en caso de requerir algún cambio, solo se trabaja sobre el nivel que lo requiera sin necesidad de afectar otras capas.

5.4.1. Capa de persistencia

Para realizar la persistencia de datos se decidió utilizar una base de datos relacional la cual fue MySQL.

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general / Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web.

5.4.2. Capa de negocio

Para la implementación de la capa de lógica de negocio se utilizó el framework Lumen³³ en su versión 7.8, el cual a su vez es un micro framework de Laravel³⁴, especialmente adaptado para el desarrollo de APIs REST. Para la gestión de dependencias fue necesario utilizar Composer v1.10.10.

Dado que esta capa debe ser capaz de acceder a la capa de persistencia, se utiliza la librería Eloquent ORM³⁵ la cual está incorporada en Lumen.

Para el envío de correos electrónicos se utilizó la librería Mail³⁶, la cual permite entre otras cosas el envío de correos electrónicos mediante una cuenta de Gmail.

Adicionalmente, la documentación de los recursos expuestos por la API REST se realizó mediante Swagger³⁷, específicamente utilizando la librería Swagger-Lume³⁸, la cual permite documentar los recursos mediante comentarios en el código PHP.

5.4.3. Capa de presentación

Esta capa cuenta con dos aplicaciones, el sitio web y la aplicación para dispositivos móviles. En ambas se utilizó el framework Angular, pero en el caso de la segunda, se la configuró para ser una PWA

5.4.4. Web

Para el desarrollo de esta capa fue utilizado el framework Angular en su versión 10.1.1, para desarrollar el sitio web como SPA (Single Page Application). Para darle estilo al sitio, se utilizó Angular Material³⁹ como framework CSS.

Se buscó que el diseño del sitio sea responsive, ya que este será accedido desde diversos dispositivos con diversas resoluciones de pantalla.

La comunicación de esta capa con la de lógica se realiza mediante el servicio REST utilizando la tecnología RXJS⁴⁰ ya integrada en Angular.

5.4.5. Aplicación PWA

Además de lo mencionado para la aplicación web, en la aplicación para dispositivos móviles se añade una librería y se configuró el proyecto para que los navegadores web la reconozcan como una aplicación PWA instalable.

5.5. Problemas encontrados

En este apartado se detallan algunos problemas presentados durante el desarrollo del proyecto y cuáles fueron las medidas para enfrentarlos.

El primer problema con el que nos encontramos al comenzar con el proyecto, fue la instalación de Docker. El problema consistió en la incapacidad para instalar la característica WSL (Windows Subsystem for Linux) en la computadora de uno de los integrantes del equipo. Luego de investigar se logró definir que el problema se origina por una incompatibilidad con la versión del sistema operativo Windows 10 Home, ya que la característica mencionada solo es compatible con las versiones Pro y Enterprise. La solución del problema fue reinstalar el sistema con una de esas versiones.

El siguiente problema se detectó durante la creación del proyecto en PHP con el framework Lumen, el cual impedía la correcta generación del mismo. El problema era causado por no tener instaladas las dependencias requeridas por el framework. Luego de varias horas de investigación se logró la instalación de las mismas, solucionando así el problema.

Otro de los inconvenientes a los que nos tuvimos que enfrentar fue la interacción con el cliente. Si bien siempre estuvo disponible para evacuar nuestras dudas, fue muy difícil entablar una negociación respecto a los requerimientos del sistema, debido a su postura poco flexible ante nuestras propuestas, incluso solicitando nuevas funcionalidades en cada oportunidad.

Durante el transcurso del proyecto nos encontramos con la necesidad de utilizar librerías o tecnologías desconocidas, y si bien esto no se considera como un problema imprevisto, sí representó un obstáculo y requirió tiempo de investigación para mitigar los riesgos.

Una de las tecnologías necesarias pero que se desconocía totalmente es Docker. Ninguno de los miembros del equipo la había utilizado previamente y se requirió de varios días de investigación y pruebas para entender qué ofrece y cómo es su funcionamiento básico.

En el caso de los frameworks, Lumen se destaca por haber sido utilizado para implementar la lógica de negocio. Si bien se tenía experiencia con el lenguaje PHP, este framework era totalmente desconocido. Antes de iniciar la implementación final, fue necesario investigar su funcionamiento y crear proyectos para realizar pruebas a pequeña escala.

Otras de las tecnologías y frameworks utilizados que se debieron investigar debido a su desconocimiento fueron: la tecnología PWA utilizada para desarrollar la aplicación móvil, el framework CSS Angular Material, la librería para documentar la API REST con Swagger, y la librería para generar un documento PDF a partir de código HTML.

5.6. Horas dedicadas

Según lo planificado al inicio del proyecto, se deberían dedicar 20 horas semanales por cada integrante del equipo, las cuales se distribuyeron en 7 horas los lunes y viernes, y 6 horas los domingos. En total, se dedicarían 320 horas por persona, sumando un total de 960 horas.

La cantidad de horas dedicadas al proyecto tuvo una desviación del 2% aproximadamente con respecto a lo planificado. En total se trabajaron 976 horas dando un promedio de 325 horas por integrante del equipo.

A continuación se muestra una tabla donde se muestran las horas dedicadas a las etapas del proyecto.

	Horas previstas	Horas dedicadas
Investigación	150	173
Documentación	160	183
Análisis y diseño	120	132
Implementación	480	448
Testing	50	40
Total	960	976

Tabla 1. Comparación por etapa de horas previstas con horas reales

5.7. El sistema desarrollado

El sistema desarrollado consta de dos componentes principales, la aplicación web y la aplicación para dispositivos móviles. A continuación se mencionan las principales características de cada una.

5.7.1. Aplicación web

La aplicación web podrá ser utilizada por todos los usuarios del sistema para realizar tareas de acuerdo al rol asignado.

A modo de ejemplo, los usuarios administradores del sistema serán capaces de registrar los cursos, áreas de estudio, tipos de curso, carreras, sedes y períodos de inscripción a cursos y exámenes, además de dar de alta nuevos usuarios.

Los funcionarios administrativos podrán realizar la asignación de docentes a cursos y exámenes, verificar las actas de los mismos y justificar inasistencias de estudiantes.

En el caso de los docentes, estos podrán realizar el control de asistencia y el ingreso de calificaciones de los cursos que dicta.

Por último, los estudiantes podrán gestionar las inscripciones a cursos y exámenes de las carreras que cursa, además de consultar su escolaridad pudiendo verla en el sitio web o descargarla en formato PDF. Las escolaridades expedidas de esta última manera cuentan con un código para que cualquier persona de manera pública pueda verificar su autenticidad.

Adicionalmente, sin la necesidad de estar registrado en el sistema, cualquier persona podrá ver el listado de las carreras dictadas por la institución, consultar información específica acerca de cada carrera y postularse en caso de quererla cursar.

A continuación, se presentan algunas capturas de pantalla del resultado logrado.

UTEC

[Inicio](#)
[Login](#)

Tecnólogo Informático

El Tecnólogo en Informática forma profesionales con las capacidades para actuar en el desarrollo, puesta en marcha, mantenimiento y administración de sistemas

[Ver más](#)

Licenciatura en ciencia y tecnología de lácteos

La Licenciatura en Ciencia y Tecnología de Lácteos (LCTL) es la primera carrera en el país que forma profesionales universitarios específicamente para la industria

[Ver más](#)

Licenciatura en Análisis Alimentario

La Licenciatura en Análisis Alimentario forma profesionales para la industria de alimentos y derivados del país con competencias para emprender proyectos de innovación y

[Ver más](#)

Tecnólogo en Manejo de sistemas de producción lechera

El Tecnólogo en Manejo de Sistemas de Producción Lechera es una carrera conjunta entre el Consejo de Educación Técnico Profesional/UTU y UTEC, orientada a formar

[Ver más](#)

Ingeniería en Mecatrónica

La carrera de Ingeniería en Mecatrónica integra la Mecánica, la Electrónica y la Informática asociadas a los procesos de automatización, con un fuerte componente

[Ver más](#)

Ingeniería en Energías Renovables

La Ingeniería en Energías Renovables se orienta a la formación de profesionales que promuevan, diseñen, implementen y administren el uso de energía limpia a partir de

[Ver más](#)

Licenciatura en tecnologías de la Información

La Licenciatura en Tecnologías de la Información tiene como áreas de formación aquellas vinculadas a las líneas de desarrollo /programación, testing e infraestructura tecnológica.

[Ver más](#)

Tecnólogo en Mecatrónica Industrial

La Carrera de Tecnólogo en Mecatrónica Industrial se imparte conjuntamente con el Consejo de Educación Técnico Profesional/UTU. Es una disciplina que integra la

[Ver más](#)

Tecnólogo en Jazz y Música creativa

El Tecnólogo forma músicos intérpretes de Jazz y otros ámbitos de su influencia, con una visión abierta del género; abordando a modo de introducción las áreas de

[Ver más](#)

Ingeniería en Sistemas de Riego, Drenaje y Manejo de efluentes

Este programa se sustenta en un Plan de Estudios orientado a la atención de la problemática del uso racional y sostenible de los recursos hídricos y tratamiento de efluentes en los

[Ver más](#)

Imagen VI. Página de inicio con listado de carreras

Tecnólogo Informático

[Postularse](#)

Descripción de la carrera

El Tecnólogo en Informática forma profesionales con las capacidades para actuar en el desarrollo, puesta en marcha, mantenimiento y administración de sistemas informáticos. Los egresados de esta carrera pueden participar como programadores y técnicos calificados en tareas de desarrollo de proyectos de variada complejidad.

Esta carrera es gestionada por UTEC en el interior del país (Maldonado, San José y Paysandú). Aquellos estudiantes que quieran anotarse a esta carrera en Montevideo deben hacerlo en este link.

Duración de la carrera

Cantidad de semestres: 6

Sedes en las que se dicta

San José de Mayo - San José

Montevideo - Montevideo

Cursos de la carrera

Nombre	Semestre	Créditos	Optativo
Principios de Programación	1	0	Si
Matemática Discreta y Lógica 1	1	12	Si
Inglés Técnico	1	8	No
Estructuras de Datos y Algoritmos	2	12	Si
Matemática Discreta y Lógica 2	2	6	Si
Bases de Datos 1	2	12	Si
Programación Avanzada C++	3	16	Si

Previaturas



Imagen VII. Vista detallada de una carrera

UTEC

Inicio Logout

Gestión de usuario

Usuarios

Gestión de carreras

Carreras

Cursos

Áreas de estudio

Tipos de curso

Otros

Año lectivo

Sedes

Mis datos

Cambiar contraseña

Gestión de carreras

Ingrese los datos

Información general

Nombre

Tecnólogo Informático

Descripción

El Tecnólogo en Informática forma profesionales con las capacidades para actuar en el desarrollo, puesta en marcha, mantenimiento y administración de sistemas informáticos. Los egresados de esta carrera pueden participar como programadores y técnicos calificados en tareas de desarrollo de proyectos de variada complejidad.

Esta carrera es gestionada por UTEC en el interior del país (Maldonado, San José y Paysandú). Aquellos estudiantes que quieran matricularse a esta carrera en Montevideo deben basarse en este link.

Sedes en las que se dicta

Seleccione una sedes

Montevideo

Agregar sede

San José

Áreas de estudio

Seleccione un área

redes y arquitectura de computadoras

Créditos

25

Agregar área de estudio

matemáticas (30 creditos)

idiomas (8 creditos)

programacion (50 creditos)

Ingrese los cursos para el semestre 3

Seleccione un curso

Bases de Datos 2

Tipo de curso

Obligatorio

Agregar curso

Confirmar semestre

Programación Avanzada C++

Semestre 1

Quitar

Matemática Discreta y Lógica 1

Inglés Técnico

Principios de Programación

Semestre 2

Quitar

Matemática Discreta y Lógica 2

Bases de Datos 1

Estructuras de Datos y Algoritmos

Volver

Acceder

Imagen VIII. Registro de carrera en el sistema

32

UTEC

Inicio Logout

Administración

Asignar docente
Revisar postulaciones
Revisar actas de cursos / exámenes
Justificar inasistencias

Mis datos

Cambiar contraseña

Revisión de actas

Lista de actas de curso

Estructuras de Datos y Algoritmos - (Curso)

Seleccionar

Estructuras de Datos y Algoritmos

ID: 46

Tipo de acta: Curso

Período: 2020-2S

Confirmada: no

Cédula	Nombre	Apellido	Nota
00000000	Juan	Alvarez	4.11
22222200	Lucas	Garrido	4.01
33333300	Sebastian	Morales	4.12
44444400	Octavio	Cepergo	4.15
55555500	Gustavo	Cerdeña	4.15
66666600	Joaquin	Suarez	4.13
77777700	Carlos	Balbani	4.57
88888800	Julio	Arrieta	4.18
99999900	Edward	Rodriguez	4.15

Volver

Confirmar acta

Imagen IX. Revisión de acta de curso o examen

33

UTEC

Inicio Logout

Área docente

Ingresar resultados de cursos

Ingresar resultados de exámenes

Control de asistencia

Mis datos

Cambiar contraseña

Control de asistencia

Lista de curso

Estructuras de Datos y Algoritmos (San José de Mayo)

Seleccionar

Cédula	Nombre	Apellido	Asistencias	Inasistencias restantes	
00000000	Juan	Alvarez	3	3	Asistió
22222200	Lucas	Garrido	2.5	2.5	Asistió
33333300	Sebastian	Morales	3	3	Llegada tarde
44444400	Octavio	Cepergo	3	3	No asistió
55555500	Gustavo	Cerdeña	2.5	2.5	Asistió
66666600	Joaquin	Suarez	3	3	Asistió
77777700	Carlos	Balbani	1.5	1.5	Asistió
88888800	Julio	Arrieta	3	3	Asistió
99999900	Edward	Rodriguez	3	3	Asistió

Volver

Confirmar

Imagen X. Control de asistencia

UTEC

Inicio

Logout

Académico

Escolaridad

Inscripción a cursos

Inscripción a exámenes

Mis datos

Cambiar contraseña

Inscripción a cursos

Lista de carreras inscripto

Tecnólogo Informático

Seleccionar

Estructuras de Datos y Algoritmos

Ya inscrito

Matemática Discreta y Lógica 2

Curso ya aprobado

Bases de Datos 1

Curso ya aprobado

Programación de Aplicaciones Java

No se cumple con las previas

Ingeniería de Software

No se cumple con las previas

Probabilidad y Estadística

Taller de Sistemas de Información .NET

Taller de Aprender a gestionar una empresa

Proyecto

Volver

Confirmar

Imagen XI. Inscripción a cursos

UTEC - Universidad Tecnológica

Certificado de escolaridad

Resultados finales e intermedios

Fecha de emisión: 31/10/2020

Cédula:	00000000
Nombre:	Juan
Apellido:	Alvarez
Carrera:	Tecnólogo Informático
Nota promedio:	4.11

Progreso académico

Semestre 1

Curso	Tipo	Período	Nota
Principios de Programación	Curso	2018-1S	4.11
Matemática Discreta y Lógica 1	Curso	2018-1S	4.11
Inglés Técnico	Curso	2018-1S	4.11

 Código de verificación: **N88P-QPDB-T1MNV4G**. (Válido hasta: **01/12/2020**)

 Puede validar esta escolaridad en <http://bedelia.utec.edu.uy/verificarEscolaridad>

Imagen XII. Escolaridad en formato PDF

5.7.2. Aplicación móvil

La aplicación móvil está diseñada exclusivamente para los estudiantes de la institución. Desde ella podrán consultar su escolaridad e inscribirse a cursos y exámenes.



Imagen XIII. Inicio

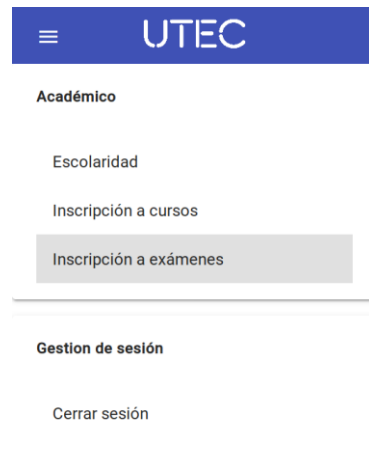


Imagen XIV. Menú

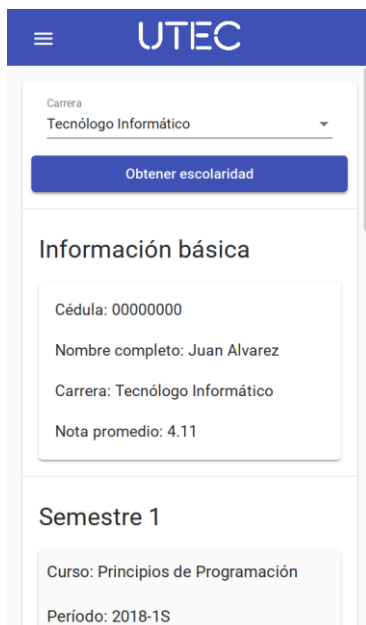


Imagen XV Consulta de escolaridad

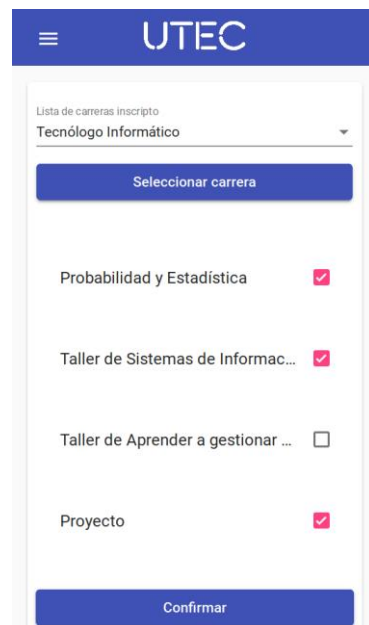


Imagen XVI Inscripción a curso

6. Conclusiones

Como conclusión se puede decir que se logró cumplir con los requerimientos funcionales como no funcionales establecidos por el cliente, cubriendo incluso el requerimiento opcional consistente en la utilización de Docker como plataforma para la infraestructura.

Se logró aplicar los conocimientos adquiridos durante la carrera y los adquiridos durante la investigación específica para este proyecto, logrando así la utilización de tecnologías que aún no conocíamos.

En lo que respecta a las dinámicas de equipo, se logró una correcta comunicación y coordinación de sus miembros gracias a la utilización de nuevas herramientas como Swagger, Trello, Toggl⁴¹ y Discord⁴². Adicionalmente consideramos que nuestra planificación y estimación de tiempo fue acertada teniendo en cuenta que por las características del proyecto había varios factores desconocidos por los integrantes del equipo.

Finalmente, podemos decir que con la realización de este proyecto hemos ganado experiencia en la manera de interactuar con el cliente, descubriendo nuestros puntos fuertes y débiles. Además se adquirió experiencia y conocimiento en tecnologías y herramientas, lo cual es de mucha ayuda en el mundo laboral actual.

7. Trabajo a futuro

Durante el transcurso del proyecto, a los diferentes miembros del equipo se le ocurrieron ideas de posibles características que sería interesante integrar al sistema, pero dado que éstas estaban por fuera de los requerimientos solicitados por el cliente, se las menciona a continuación como trabajo a futuro.

Mejora en casos de uso: Como parte de las funcionalidades que el sistema ofrece, sería de utilidad la adición de las siguientes funcionalidades:

- Copiar año lectivo: Lo cual permitiría a los administradores del sistema copiar las fechas de inicio y fin de los períodos de un año lectivo anterior.
- Verificar escolaridad: Para aumentar la seguridad de la información de los estudiantes, sería recomendable incorporar una verificación por CAPTCHA⁴³ al momento de intentar verificar un código de escolaridad.
- Notificaciones: Una nueva funcionalidad que permitiría a los estudiantes recibir notificaciones con la nota obtenida en los cursos y exámenes a los que se encuentra inscrito.

Mejora del responsive: Si bien el sitio desarrollado es responsive, existen algunos componentes que no respetan este patrón de diseño. Este es el caso de las tablas, las cuales deberían reorganizar su distribución al ser visualizadas en un dispositivo móvil, para permitir su fácil interpretación.

Publicación de la aplicación en tiendas: Para facilitar el acceso a la aplicación desde los dispositivos móviles, sería de utilidad que la misma se encuentre publicada en las tiendas de aplicaciones de Android e IOS. Esto puede ser logrado de manera sencilla utilizando la tecnología TWA⁴⁴.

Correos electrónicos: Para el envío de correos electrónicos sería conveniente la utilización de un servidor SMTP⁴⁵ propio o integrar algún servicio de terceros mediante el cual se puedan enviar correos utilizando su API (ej., SendInBlue⁴⁶).

Mejora estética: Para mejorar el aspecto tanto del sitio web como de la aplicación móvil sería conveniente la utilización de recursos gráficos como lo son los íconos y también la incorporación de un tema oscuro. Ambas características mejoran la experiencia de los usuarios incrementando la amigabilidad y la usabilidad.

8. Referencias

- ¹ Web. Delgado, Hugo. (2015). Recuperado 01 de diciembre, 2020, de <https://disenowebakus.net/world-wide-web-www.php>
- ² HTML. Código Facilito. Recuperado 01 de diciembre, 2020, de <https://codigofacilito.com/articulos/que-es-html>
- ³ CSS. Websarrolladores. Recuperado 01 de diciembre, 2020 de <https://websarrolladores.com/2018/08/30/css/>
- ⁴ JavaScript. Velneo. Recuperado 01 de diciembre, 2020 de <https://velneo.es/que-es-javascript/>
- ⁵ Framework. Metodología. Recuperado 01 de diciembre, 2020 de <https://metodologia.es/marco-de-trabajo/>
- ⁶ TypeScript. Wikipedia. Recuperado 01 de diciembre, 2020 de <https://es.wikipedia.org/wiki/TypeScript>
- ⁷ PHP. Ryte Wiki. Recuperado 01 de diciembre, 2020 de <https://es.ryte.com/wiki/PHP>
- ⁸ Angular. Uдеми. Recuperado 01 de diciembre, 2020 de <https://www.udemy.com/course/angular-framework-para-aplicaciones-web/>
- ⁹ RESTful. Wikipedia. Recuperado 01 de diciembre, 2020 de https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional
- ¹⁰ PWA. Wikipedia. Recuperado 02 de diciembre, 2020 de https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web_progresiva
- ¹¹ Arquitectura de software. SG. Recuperado 02 de diciembre, 2020 de <https://sg.com.mx/revista/27/arquitectura-software>
- ¹² Modelo cliente-servidor. InfraNetworking. Recuperado 03 de diciembre, 2020 de <https://blog.infranetworking.com/modelo-cliente-servidor/>
- ¹³ Arquitectura en capas. EcuRed. Recuperado 03 de diciembre, 2020 de https://www.ecured.cu/Arquitectura_en_Capas#Arquitectura_en_Tres_Capas
- ¹⁴ Mobile. La Comisión Federal de Comercio. Recuperado 03 de diciembre, 2020 de <https://www.consumidor.ftc.gov/articulos/s0018-aplicaciones-moviles-que-son-y-como-funcionan#Nociones>
- ¹⁵ Android. Culturación. Recuperado 03 de diciembre, 2020 de <https://culturacion.com/android-principales-caracteristicas-del-sistema-operativo-de-google/>

- ¹⁶ iOS. Culturacion. Recuperado 03 de diciembre, 2020 de <https://culturacion.com/ios-el-sistema-operativo-movil-de-apple/>
- ¹⁷ Modelo de vistas de arquitectura 4+1. Philippe Kruchten. Recuperado 03 de diciembre, 2020 de http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=materias:modelo4_1.pdf
- ¹⁸ Trello. Trello. Recuperado 04 de diciembre, 2020 de <https://trello.com/es>
- ¹⁹ Frontend. Platzi. Recuperado 04 de diciembre, 2020 de <https://platzi.com/blog/que-es-frontend-y-backend/>
- ²⁰ Backend. Platzi. Recuperado 04 de diciembre, 2020 de <https://platzi.com/blog/que-es-backend-y-frontend/>
- ²¹ Windows 10. Microsoft. Recuperado 04 de diciembre, 2020 de <https://www.microsoft.com/es-mx/windows/>
- ²² Kubuntu 20.04. Kubuntu. Recuperado 04 de diciembre, 2020 de <https://kubuntu.org/>
- ²³ Visual Studio Code. Visual Studio Code. Recuperado 04 de diciembre, 2020 de <https://code.visualstudio.com/>
- ²⁴ Composer. Composer. Recuperado 04 de diciembre, 2020 de <https://getcomposer.org/>
- ²⁵ NodeJS. NodeJS. Recuperado 04 de diciembre, 2020 de <https://nodejs.org/>
- ²⁶ NPM. NPM. Recuperado 04 de diciembre, 2020 de <https://www.npmjs.com/>
- ²⁷ Angular CLI. Angular CLI. Recuperado 04 de diciembre, 2020 de <https://cli.angular.io/>
- ²⁸ Docker. Docker. Recuperado 04 de diciembre, 2020 de <https://www.docker.com/>
- ²⁹ MySQL. MySQL. Recuperado 04 de diciembre, 2020 de <https://www.mysql.com/>
- ³⁰ phpMyAdmin. phpMyAdmin. Recuperado 04 de diciembre, 07 de <https://www.phpmyadmin.net/>
- ³¹ Git. Git. Recuperado 07 de diciembre, 2020 de <https://git-scm.com/>
- ³² GitHub. GitHub. Recuperado 07 de diciembre, 2020 de <https://github.com/>
- ³³ Lumen. Lumen. Recuperado 07 de diciembre, 2020 de <https://lumen.laravel.com/>
- ³⁴ Laravel. Laravel. Recuperado 07 de diciembre, 2020 de <https://laravel.com/>
- ³⁵ Eloquent ORM. Laravel. Recuperado 10 de diciembre, 2020 de <https://laravel.com/docs/5.0/eloquent>
- ³⁶ Mail. Laravel. Recuperado 10 de diciembre, 2020 de <https://laravel.com/docs/8.x/mail>

- ³⁷ Swagger. Swagger. Recuperado 10 de diciembre, 2020 de <https://swagger.io/>
- ³⁸ Swagger-Lume. DarkaOnLine GitHub. Recuperado 10 de diciembre, 2020 de <https://github.com/DarkaOnLine/SwaggerLume>
- ³⁹ Angular Material. Material. Recuperado 10 de diciembre, 2020 de <https://material.angular.io/>
- ⁴⁰ RXJS. Angular. Recuperado 10 de diciembre, 2020 de <https://angular.io/guide/rx-library>
- ⁴¹ Toggl. Toggl. Recuperado 10 de diciembre, 2020 de <https://toggl.com/>
- ⁴² Discord. Discord. Recuperado 10 de diciembre, 2020 de <https://discord.com/>
- ⁴³ CAPTCHA. Universidad de Alicante. Recuperado 10 de diciembre, 2020 de <http://accesibilidadweb.dlsi.ua.es/?menu=que-es-un-captcha-problemas-accesibilidad>
- ⁴⁴ TWA. Google. Recuperado 10 de diciembre, 2020 de <https://developers.google.com/web/android/trusted-web-activity>
- ⁴⁵ SMTP. Mailjet. Recuperado 10 de diciembre, 2020 de <https://es.mailjet.com/blog/news/servidor-smtp/>
- ⁴⁶ SendInBlue. SendInBlue. Recuperado 10 de diciembre, 2020 de <https://es.sendinblue.com/>