

A.B.E.P APLICATIVO DE BÚSQUEDA DE EMPLEOS Y PRACTICA

CARLOS EDUARDO MORALES PINZÓN

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA

FACULTAD INGENIERÍA DE SISTEMAS

TECNOLOGÍA Y SOCIEDAD

BUCARAMANGA

2021

A.B.E.P APLICATIVO DE BÚSQUEDA DE EMPLEOS Y PRACTICA

CARLOS EDUARDO MORALES PINZÓN

PROYECTO DE GRADO PRESENTADO PARA OPTAR AL TÍTULO DE
INGENIERO DE SISTEMAS

DIRECTOR

JUAN SEBASTIÁN CÁRDENAS ARENAS
INGENIERO DE SISTEMAS

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
FACULTAD INGENIERÍA DE SISTEMAS
TECNOLOGÍA Y SOCIEDAD
BUCARAMANGA

2021

CONTENIDO

RESUMEN	1
INTRODUCCIÓN	2
PLANTEAMIENTO DEL PROBLEMA	2
JUSTIFICACIÓN	4
OBJETIVOS	5
SOLUCIÓN	6
ESTADO DEL ARTE	7
MARCO CONCEPTUAL	9
METODOLOGÍA	13
DESARROLLO DEL PROYECTO	15
RESULTADOS	27
CONCLUSIONES	28
RECOMENDACIONES O TRABAJOS FUTUROS	29
REFERENCIAS BIBLIOGRÁFICAS	30
ANEXOS	32

LISTA DE ILUSTRACIONES

Ilustración 1: Medios usados para la búsqueda de prácticas	2
Ilustración 2: Obstáculos para encontrar trabajo	3
Ilustración 3: Funcionamiento de una REST API	10
Ilustración 4: Conexión a una base de datos	10
Ilustración 5: Funcionamiento de un Modelo-Vista-Controlador	11
Ilustración 6: Funcionamiento de un Modelo-Vista-Template	12
Ilustración 7: Diagrama de Casos de Uso - Login	16
Ilustración 8: Diagrama de Casos de Uso	17
Ilustración 9: Diagrama Entidad-Relación	18
Ilustración 10: Demostración de Mockups	19
Ilustración 11: Diagrama de Arquitectura	19
Ilustración 12: Demostración de un modelo de BD	20
Ilustración 13: Jerarquía y funcionamiento MTV	21
Ilustración 14: Demostración de una View	21
Ilustración 15: Demostración de la plantilla	22
Ilustración 16: Implementación rest framework	22
Ilustración 17: Demostración del serializer y el view del API	23
Ilustración 18: Jerarquía y un layout	24
Ilustración 19: Implementación de una llamada usando Retrofit	25
Ilustración 20: Demostración de UploadFile	26

LISTA DE TABLAS

Tabla 1: Tabla comparativa de soluciones al proyecto.....	6
Tabla 2: Estado del Arte	7
Tabla 3: Metodología en Detalle	14
Tabla 4: Resumen de Comparación en Pruebas.....	26

LISTA DE ANEXOS

Anexo A: Documento basado en cuadros comparativos que investiga aplicaciones similares al proyecto	32
Anexo B: Mockups del Aplicativo Web	36
Anexo C: Mockups del Aplicativo Móvil	39
Anexo D: Documentación de pruebas de experiencia de usuario	40

RESUMEN

La difusión de la tecnología en la sociedad ha provocado una revolución en el funcionamiento del mundo profesional. Este artículo presenta una solución que utiliza la implementación de software en dispositivos móviles Android y una aplicación web que conecta a los pasantes y recién graduados sin experiencia con las empresas. Esto permite que tanto los pasantes como los recién graduados obtengan un trabajo en el que puedan acumular experiencia y conocimientos para comenzar a competir en el mundo profesional. Esta solución permite a las empresas publicar ofertas de trabajo y tener un control de los postulantes y su currículum a través de una aplicación web. Los usuarios pueden crear fácilmente una cuenta para encontrar sus mejores opciones y postularse a través de la aplicación web o móvil.

Este proyecto se realizará dividiendo ambas aplicaciones en ciclos de trabajo en el que irán organizando las funcionalidades de los aplicativos por partes. Ambas aplicaciones se dan por terminadas una vez todas sus funcionalidades han sido enlazadas.

Los resultados del proyecto será una plataforma funcional compuesta de un aplicativo web principal que maneja los usuarios y un prototipo de aplicación móvil para uso específico por los estudiantes, a diferencia de la aplicación web que ofrece servicios tanto a los estudiantes como a las empresas. La plataforma estará conectada en tiempo real a una base de datos.

Las conclusiones serán informativas para demostrar el efecto positivo del proyecto, tanto en el factor estudiantil al adquirir conocimiento aplicado como en la utilidad que representa la aplicación demostrando que puede ser útil en más áreas o proyectos con un mayor alcance.

Palabras Clave: platforms, jobs, mobile application, web application, APIs

Línea de Investigación: Tecnología y Sociedad.

INTRODUCCIÓN

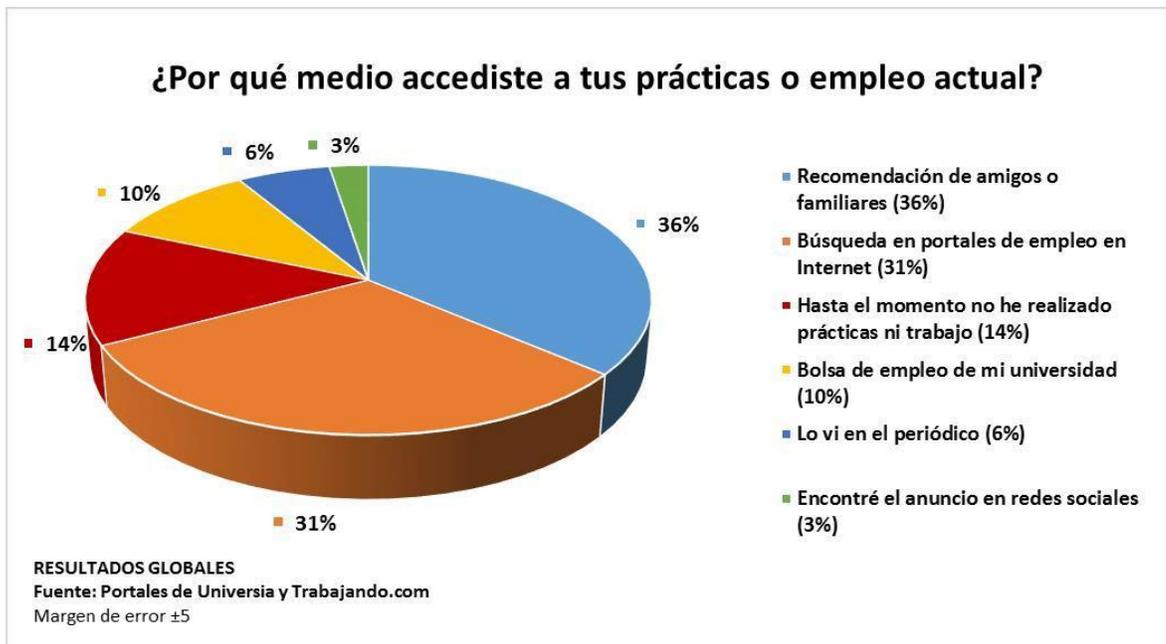
PLANTEAMIENTO DEL PROBLEMA

La humanidad avanza exponencialmente, a un ritmo acelerado y los requisitos mínimos se van volviendo mayores con el tiempo, uno de los problemas ligados que genera esto es la dificultad a la hora de poder encontrar prácticas aptas para los estudiantes o empleo para los recién graduados, ya sea por la ubicación, puesto, horario, tiempo contratado, requerimientos especiales, el extra de recibir algún pago o no y sobre todo la falta de experiencia.

De tantas opciones es posible que no todas fueron tomadas en cuenta por la falta de herramientas o métodos para dar a conocer todas las oportunidades al alcance de los estudiantes.

En la Ilustración 1 se puede apreciar una encuesta realizada por (trabajando.es, 2014) a gente de España y sur américa de habla hispana, en cual podemos apreciar que las búsquedas de empleo por medios electrónicos no dominan todavía las preferencias de búsqueda hasta la fecha citada.

Ilustración 1: Medios usados para la búsqueda de prácticas

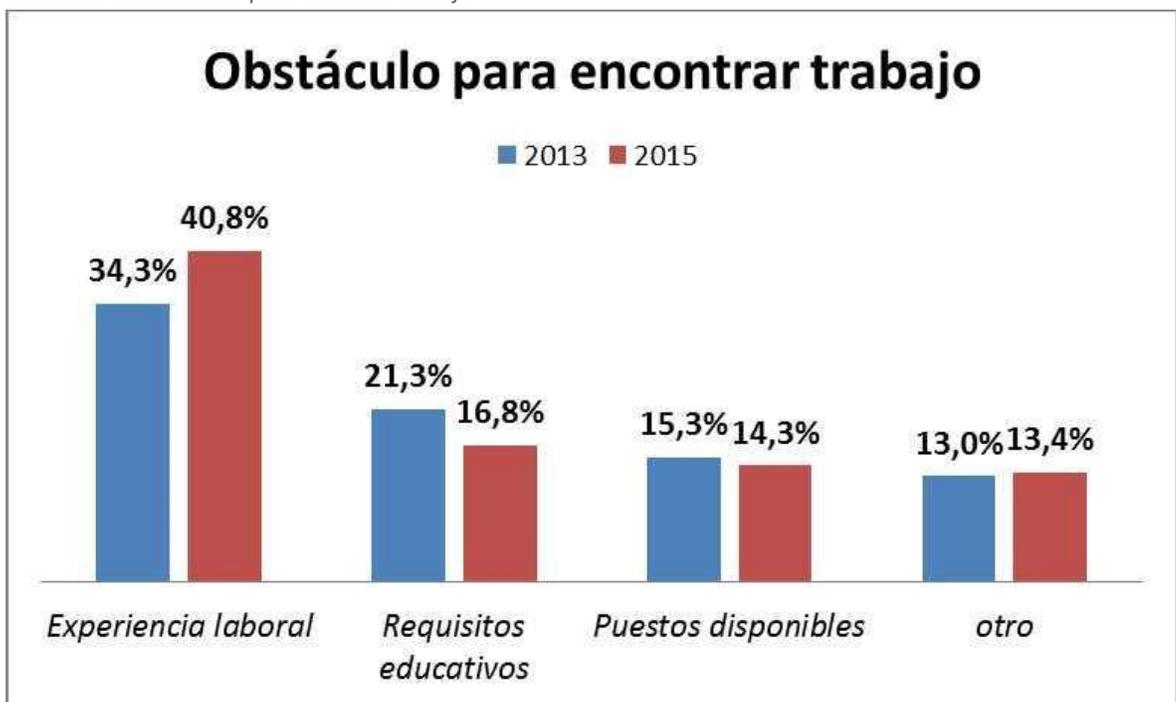


Recuperado de Trabajando. (2014). El 75% de jóvenes iberoamericanos recurre primero a portales de empleo para buscar trabajo. Recuperado de <https://www.trabajando.es/contenido/noticia/3260/EI-75--de-jovenes-iberoamericanos-recurre-primero-a-portales-de-empleo-para-buscar-trabajo.html>

La falta de experiencia es un factor muy obvio para un practicante, también lo es para un profesional recién graduado que está en busca de su primer empleo a nivel profesional en su área laboral, es común que las empresas decidan disminuir los requisitos que requiere una persona para un puesto (grado, estudios), pero aumentan sus requisitos en experiencia laboral buscando equilibrar los cambios realizados. Por ende, la población menos preparada puede aspirar a puestos mejores con mayor probabilidad que un recién graduado. Esto conlleva a que personas con menos preparación puedan optar por mejores puestos dado que tienen experiencia laboral previa de tipo técnico.

Los técnicos en promedio tienen alguna experiencia acumulada a diferencia de los profesionales universitarios, la nueva metodología planteada de varias empresas colombianas ahora dificulta más la entrada laboral de los recién graduados y de alguna forma los practicantes también. Como se puede ver en la encuesta realizada por la revista (Dinero, 2016), ver Ilustración 2.

Ilustración 2: Obstáculos para encontrar trabajo



Recuperado de Dinero. (2016). El 40,8% de los jóvenes colombianos dice no conseguir trabajo por falta de experiencia. Recuperado de <http://www.dinero.com/economia/articulo/las-expectativas-y-condiciones-laborales-de-los-jovenes-en-colombia/222244>

JUSTIFICACIÓN

- Los usuarios y las empresas ignoran los avances tecnológicos para darse a conocer
- Las empresas no pueden encontrar personal con los requisitos que necesitan
- Los usuarios no se informan de posibles ofertas que sean aptas para ellos
- Necesidad por incentivar el interés de las empresas por contratar personal sin experiencia

OBJETIVOS

Objetivo General

Desarrollar una plataforma de software compuesta por un prototipo de aplicación móvil y una página web interconectadas que facilite la búsqueda de prácticas universitarias tanto para los estudiantes como para las empresas en busca de ellos.

Objetivos Específicos

1. Realizar una investigación de aplicaciones web/móviles relacionadas con el fin de encontrar las tecnologías más adecuadas para el desarrollo del proyecto.
2. Desarrollar el aplicativo web conjunto con la base de datos que implementará para facilitar la búsqueda de prácticas universitarias.
3. Desarrollar un prototipo de aplicación móvil, el cual permita a los usuarios estudiantes consultar las ofertas existentes en la plataforma para su posterior postulación.
4. Realizar escenarios de prueba para comprobar la calidad, funcionalidad y experiencia de usuario de la plataforma.

SOLUCIÓN

A.B.E.P una plataforma para la búsqueda exclusiva de prácticas universitarias y primer empleo donde los usuarios fácilmente pueden aplicar a una oferta de su agrado y las empresas conocerlos.

A continuación, una tabla que compara algunos problemas comunes de otras aplicaciones con nuestras soluciones del proyecto:

Tabla 1: Tabla comparativa de soluciones al proyecto

Apps	Problema	Solución del proyecto
Firstjob.me	No tiene App Móvil	Se realizará un prototipo de aplicación móvil
First Job	Únicamente disponible en inglés	Será desarrollada en español, dependiendo de su lanzamiento y éxito se consideraría su traducción para un nuevo público
JobToday, First Job, Primerempleo	Disponibles únicamente para algunas zonas o países	Planeada para ser accesible a todas las empresas en todo lugar que deseen publicar
Jobandtalent, Dream Jobs	Falta de empresas asociadas al servicio en varias zonas donde lo prestan	Planeada para ser accesible y darse a conocer a todas las empresas
Computrabajo, LinkedIn	Las ofertas publicadas suelen pedir más requisitos de los necesarios al ser también servicios planeados para todo tipo de trabajo al público	Al únicamente enfocarnos en prácticas y primer empleo, solicitar requisitos aptos y lógicos para este tipo de ofertas por parte de las empresas

ESTADO DEL ARTE

Se realizó la siguiente tabla para guardar y comparar los datos encontrados de aplicaciones móviles y web con objetivos similares al de este proyecto, con el fin de recopilar datos importantes que nos muestren una idea de que tanta variedad y cuanto uso se le da a este tipo de aplicaciones.

Tabla 2: Estado del Arte

Nombre	Año Fundación	App Web	App Móvil
Primerempleo.com	2002	Una página que ofrece búsqueda de primer empleo, búsqueda de trabajo en general, cursos y publica contenido o información relevante respecto al trabajo.	Aplicación funcional desde el 2015, permite crear perfil del usuario, ver ofertas y solicitar aplicar a alguna, se enfoca en España y México.
Job Today	2014	Página para buscar trabajo en España	Aplicación para empresas para buscar y contratar empleados en España.
First Job	2012	-	Aplicación en Ingles que funciona buscando empleos principalmente en India.
Firstjob.me	2012	Web de origen chileno que sirve para buscar empleos, se ha ampliado a varios países y cuenta con servicio de Video entrevista Online.	-
Jobandtalent empleo y trabajo	2008	Una página informativa y de soporte acerca de la aplicación	Una aplicación para buscar trabajo en general con bastantes opciones y un método resumido de conseguir trabajo basándose únicamente en datos simples e importantes, carece de empresas suficientes para
CompuTrabajo	1999	Una de las páginas más famosas en	Una aplicación con las mismas utilidades y funciones

		Colombia para búsqueda de empleo. Fundada y creada en Escocia	de su página carece de empleos sin experiencia y las ofertas suelen ser más exigentes.
Dream Jobs	2016	-	Una aplicación muy similar a jobandtalent, usa una funcionalidad de GPS para mostrar ofertas cercanas a la ubicación y así poder enviar hoja de vida, igualmente carece de ofertas y empresas asociadas para su funcionalidad
LinkedIn Job Search	2002	Como un derivado de LinkedIn, red social laboral está en una de sus opciones, Jobs, la alternativa para búsqueda de trabajo entre las compañías y sus usuarios registrados a nivel Mundial.	Se sincroniza con la red social laboral de LinkedIn, dando muchas opciones al usuario para buscar trabajo, tiene la opción para búsqueda de empleos sin experiencia y prácticas, es necesario tener una cuenta del perfil social LinkedIn, la mayoría de ofertas son más exigentes.

MARCO CONCEPTUAL

En el marco teórico de este documento se presentan los términos más importantes que se van a usar dentro del proyecto. Estos términos son importantes dentro del desarrollo de aplicaciones móviles y web.

Los términos clave tomados fueron:

- Aplicación Móvil
- Aplicación Web
- Desarrollo web
- Front End y Back End
- Base de datos
- Práctica
- API
- MTV

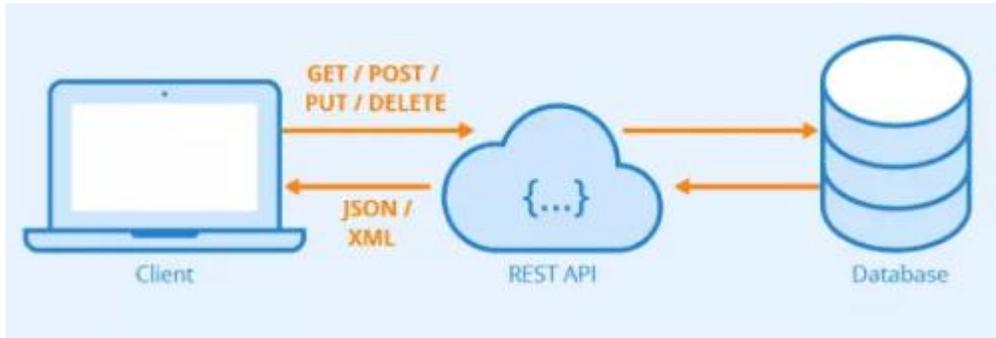
Práctica: La Práctica Profesional es el proceso mediante el cual los alumnos de un Establecimiento Universitario formalizan las habilidades y conocimientos adquiridos durante su formación profesional, entendiendo como un período que posibilita validar, en un contexto laboral, los aprendizajes obtenidos en la etapa formativa. Además, representa la vinculación del Establecimiento con el Mundo Empresarial. (Municipalidad Papudo, 2002)

Aplicación Web: Es un programa aplicativo que está almacenado en un servidor remoto y es entregado a través de internet por medio de una interfaz del navegador. Según Jarel Remick editor de Web.AppStorm, cualquier componente de sitio web que ejecute alguna función para el usuario califica como Aplicativo Web. (Rouse, Web App, 2011)

Aplicación Móvil: La influencia de las aplicaciones móviles no ha parado de crecer y en los últimos años se han convertido en una industria muy grande, pero primero que todo debemos definir las aplicaciones móviles. Las aplicaciones móviles según el libro Empowering Learners With Mobile Open – Access Learning Initiatives “son las que permiten a las tabletas y teléfonos celulares funcionar. Se ha vuelto prácticamente imposible usar alguna función del celular sin usar una aplicación que ha sido descargada de algún store. Ya sea navegar la web, jugar, o buscar información sobre salud, es probablemente que esté hecha usando una aplicación móvil”. (Mills, Wake, 2017)

API: Las APIs actualmente son muy importantes dentro del desarrollo de aplicaciones en Android. Jorgensen (2000) nos dice que un API es una colección de funciones de software y procedimientos, las cuales pueden ser ejecutadas por otras aplicaciones software. Los desarrolladores codifican llamados a APIs existentes para usar su funcionalidad. Estos llamados no se notan y para el usuario final son como si fueran inexistentes. (Naeem, 2021)

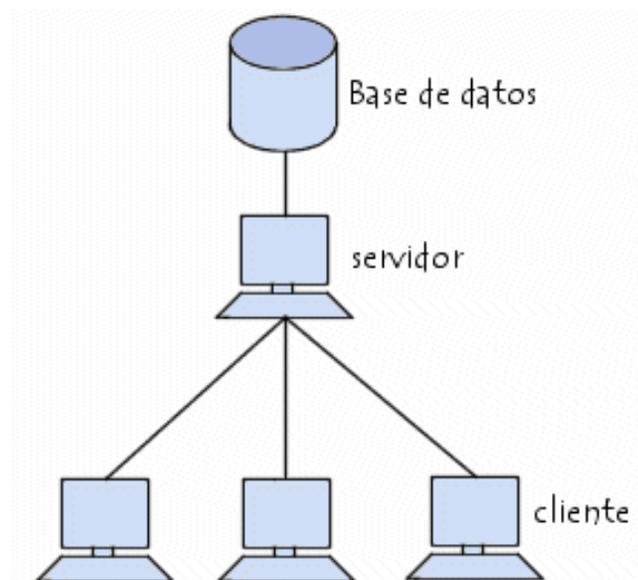
Ilustración 3: Funcionamiento de una REST API



Recuperado de <https://www.astera.com/es/tipo/blog/definici%C3%B3n-de-la-API-de-descanso/>

Base de datos: Es una colección de información organizada a la que se puede acceder, modificar y actualizar fácilmente. Una base de datos relacional es tabular en donde sus datos están definidos, pueden ser accesibles y reorganizados de varias maneras diferentes. (Rouse, database (DB), 2017)

Ilustración 4: Conexión a una base de datos



Recuperado de <https://sites.google.com/site/proyectopaquetes/capitulo-4-manejo-de-base-de-datos-y-analisis-de-la-informacion>

Front End: Se puede denominar al front end como el conjunto que conforma la interfaz del aplicativo web o móvil. Según Iván Alvarado editor de Serprogramador.es, El front end son todas aquellas tecnologías que corren del lado del cliente, es decir, todas aquellas tecnologías que corren del lado del navegador web, generalizando más que nada en tres lenguajes, HTML, CSS Y JavaScript. (Diaz, 2014)

Back End: Se determina como el control de datos detrás de un aplicativo. Según Iván Alvarado editor de Serprogramador.es, El Back End emplea lenguajes como PHP, Python, .Net, Java, etc., se encarga de interactuar con bases de datos, verificar manejo de sesiones de usuarios, montar la página en un servidor, y desde éste “servidor” todas las vistas que el Front End crea, es decir, el back end realiza principalmente la manipulación de los datos. (Diaz, 2014)

Android Studio: Es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan tu productividad durante la compilación de apps para Android. (Android Studio, n.d.)

MVC: Modelo Vista Controlador es un patrón de arquitectura de software que separa los datos de nuestra aplicación de la lógica y de la interfaz de usuario obteniendo así tres componentes relacionados entre sí. (Guerrero, 2017)

Ilustración 5: Funcionamiento de un Modelo-Vista-Controlador

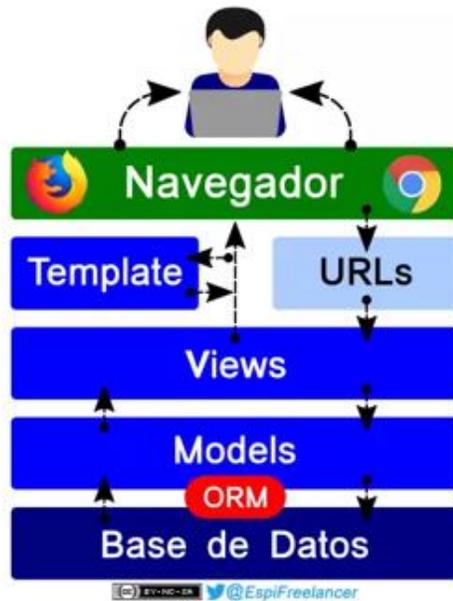


Recuperado de <http://programaenlinea.net/modelo-vista-controlador-en-php/>

MTV: Model-Template-View, el patrón MTV es muy similar al MVC. Realmente este no se desvía demasiado del patrón Modelo Vista Controlador, simplemente lo

implementa de una manera distinta y para evitar confusiones es llamado MTV. (EspiFreelancer, n.d.)

Ilustración 6: Funcionamiento de un Modelo-Vista-Template



Recuperado de <https://espifreelancer.com/mtv-django.html>

SQLite: SQLite es una biblioteca en lenguaje C que implementa un motor de base de datos SQL pequeño, rápido, autónomo, de alta confiabilidad y con todas las funciones. SQLite es el motor de base de datos más utilizado del mundo. SQLite está integrado en todos los teléfonos móviles y la mayoría de las computadoras y viene incluido dentro de innumerables aplicaciones que la gente usa todos los días. (SQLite, n.d.)

Django: Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, soporta multiplataforma al estar escrito en Python. (MDN contributors, 2020)

METODOLOGÍA

Metodología de desarrollo de software (Modelo Cascada): El modelo de la cascada, a veces llamado ciclo de vida clásico sugiere un enfoque sistemático y secuencial para el desarrollo del software, que comienza con la especificación de los requerimientos por parte del cliente y avanza a través de planeación, modelado, construcción y despliegue, para concluir con el apoyo del software terminado. (Pressman, 2010)

La metodología se encuentra enmarcada en cinco fases:

- Análisis
- Diseño
- Desarrollo
- Pruebas de funcionamiento
- Entrega

Análisis: En esta fase se analizan los requerimientos del producto a desarrollar para el cliente. Se desea definir las características del producto y esto se hace en tres tareas:

- Obtener los requerimientos
- Clasificar los requerimientos
- Personalizar el servicio

Diseño: En esta fase se desea plasmar el pensamiento de la solución mediante diagramas o esquemas considerando las mejores alternativas al solucionar el problema. A esta fase es la que se retorna si no se obtiene lo deseado en la fase de pruebas de funcionamiento. En esta fase se realizan cuatro tareas:

- Definir el escenario
- Estructurar el software
- Definir tiempos
- Asignar Recursos

Desarrollo: En esta fase se implementa el diseño en el software y se realizan las siguientes tareas:

- Codificar
- Pruebas unitarias
- Documentar el código
- Codificar ayudas

Pruebas de funcionamiento: En esta fase se verifica el funcionamiento de la aplicación en diferentes escenarios y condiciones, para eso se deben realizar las siguientes tareas:

- Emulación y simulación
- Dispositivos Reales

Entrega: Una vez terminada la depuración de la aplicación y atendido todos los requerimientos de última hora del cliente se puede dar por terminada la aplicación y se procede a la entrega del ejecutable, el código fuente, la documentación y el manual del sistema. (Gasca, Camargo, Medina, 2014)

Tabla 3: Metodología en Detalle

Objetivo específico	Parte 1	Parte 2	Parte 3
1	Investigar información acerca de Aplicaciones web y Móviles similares	Realizar un cuadro comparativo entre las aplicaciones investigadas	Realizar un cuadro de problemas en el cual se muestra la posible solución que este proyecto aporta
2	Planear el modelo y estructura de la base de datos de la plataforma	Desarrollar el aplicativo web en base a la estructura de la base de datos	Conectar el aplicativo web a la base de datos y realizar un REST api
3	Planear el modelo e interfaces que usará el prototipo de aplicativo móvil	Realizar el prototipo de aplicación móvil	Implementar la conexión al REST api
4	Realizar pruebas funcionales de ambos aplicativos en dispositivos reales	Realizar pruebas de usuario para determinar si la experiencia de usuario es agradable	Describir la experiencia en la prueba de los aplicativos

DESARROLLO DEL PROYECTO

1. Para el inicio del proyecto, se requirió una investigación de aplicaciones similares al proyecto. En el estado del arte y Anexo A, se ha adjuntado la documentación de la investigación realizada.

Se encontraron aplicaciones como:

Apps	Problema	Solución del proyecto
Firstjob.me	No tiene App Móvil	Se realizará un prototipo de aplicación móvil
First Job	Únicamente disponible en inglés	Será desarrollada en español, dependiendo de su lanzamiento y éxito se consideraría su traducción para un nuevo público
JobToday, First Job, Primerempleo	Disponible únicamente para algunas zonas o países	Planeada para ser accesible a todas las empresas en todo lugar que deseen publicar

En el Anexo A también se encuentra la documentación de la comparación entre Frameworks y lenguajes de programación seleccionados para la realización del proyecto.

A partir de este anexo se tomó que la mejor elección para el proyecto era programarlo en Django (Python) para el aplicativo web y Android Studio (Kotlin) para el aplicativo móvil.

Teniendo en cuenta la problemática del proyecto se decide que la plataforma debe contar con las siguientes características:

Aplicativo Web:

- Permite logear usuarios y empresas
- Editar perfil usuarios y empresas
- Publicar ofertas desde empresa
- Revisar aplicantes desde empresa
- Buscar y aplicar a ofertas desde usuario
- Encontrar ofertas con filtros de búsqueda desde usuario
- Visualizar ofertas ya aplicadas desde usuario

Aplicativo Móvil:

- Permite logear usuarios
- Editar perfil usuarios
- Buscar y aplicar a ofertas desde usuario
- Encontrar ofertas con filtros de búsqueda desde usuario

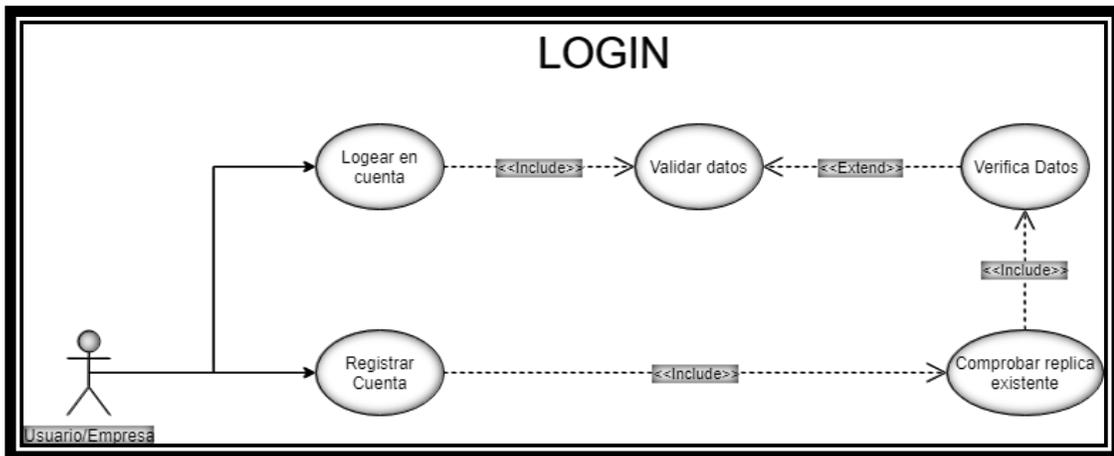
2. Se empieza con la planeación de la estructura interna de la plataforma, se diseña el diagrama de casos de uso de la aplicación.

Los actores identificados son:

Usuario: Es el estudiante o recién graduado interesado en buscar una práctica

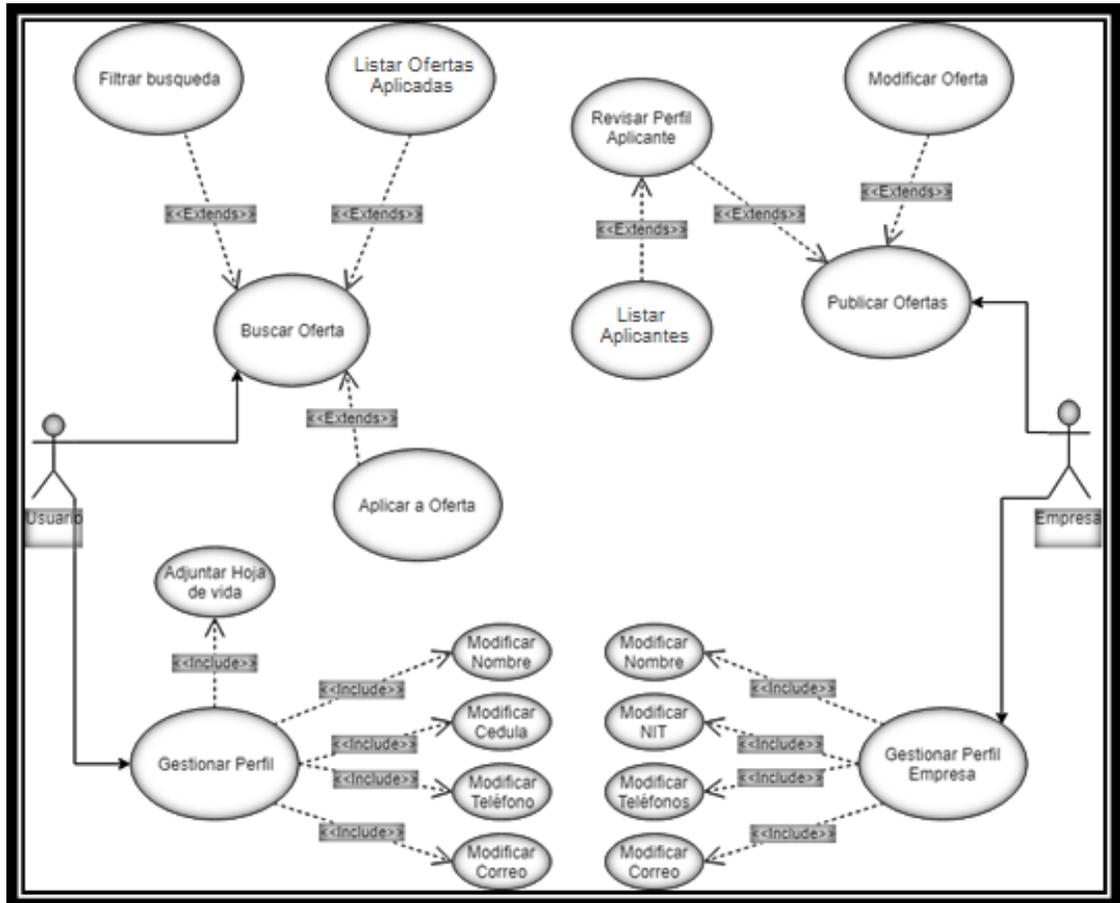
Empresa: Es la organización interesada en contratar practicantes

Ilustración 7: Diagrama de Casos de Uso - Login



Fuente: Autor

Ilustración 8: Diagrama de Casos de Uso



Fuente: Autor

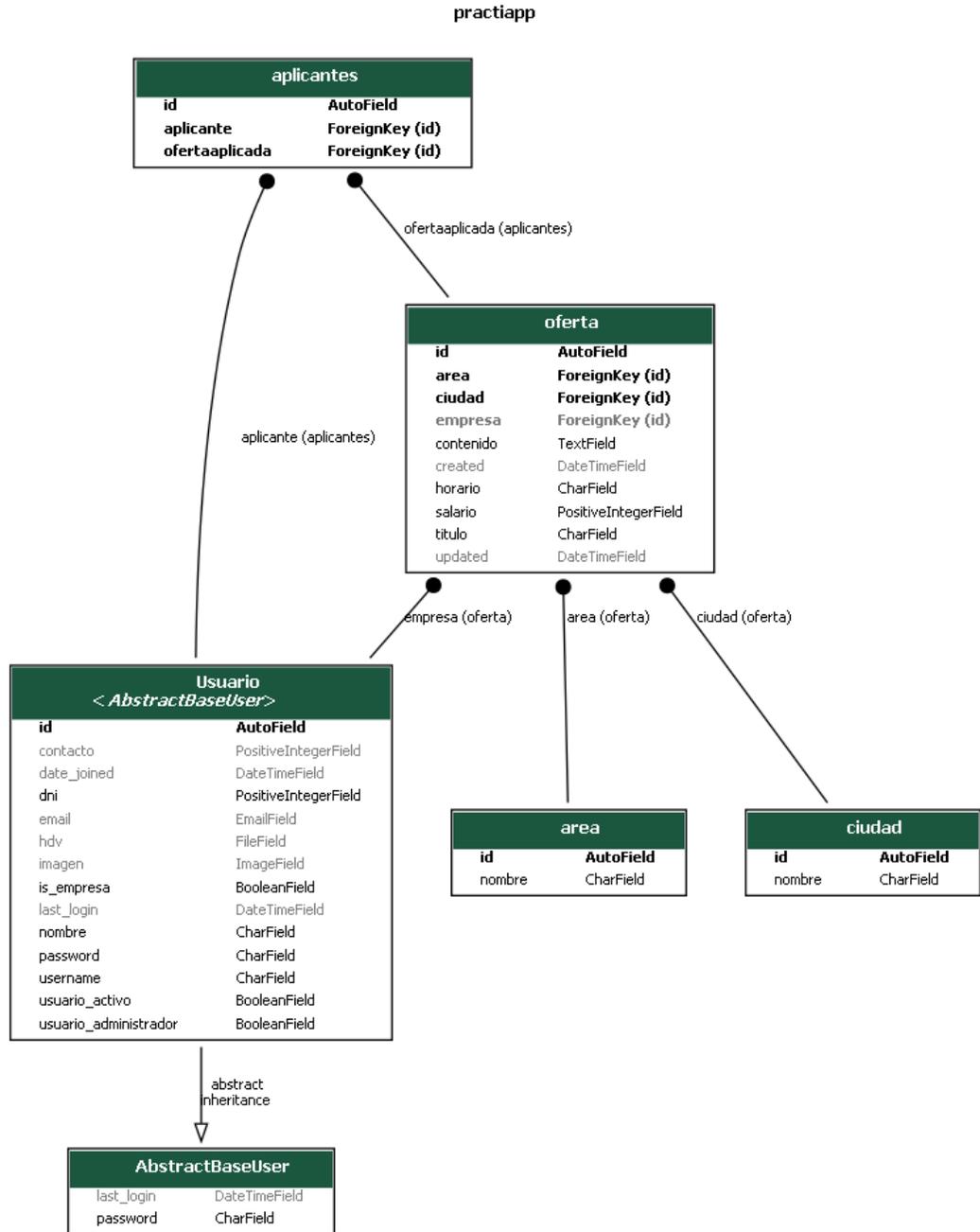
3. Se realiza el diagrama Entidad-Relación que manejará la base de datos.

El diagrama entidad-relación formado por 2 principales clases, usuario y oferta, en usuario se guardan todos los datos de los usuarios o empresas registradas.

Oferta guarda toda la información referente a las publicaciones realizadas por las empresas, llamando como llave foránea al usuario que la crea (empresa) y 2 tablas más, que guardan específicamente las áreas de conocimiento y ciudades de Colombia para seleccionar alguna de forma específica entre la lista.

La clase de aplicantes es una tabla formada solo por 2 llaves foráneas, los id de cada oferta y usuario que ha aplicado a la oferta para registrar los datos y se tenga la información de que usuarios aplicaron a que oferta.

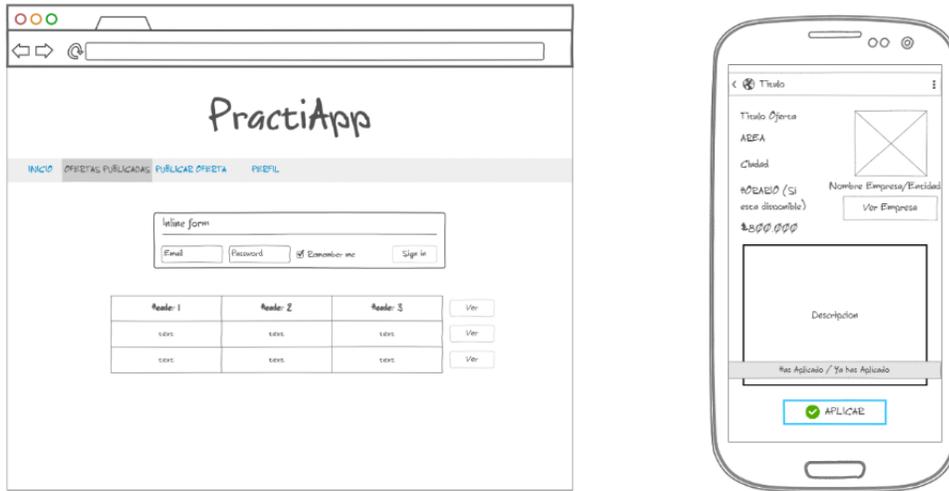
Ilustración 9: Diagrama Entidad-Relación



Fuente: Autor

- Con el funcionamiento interno de la aplicación se procede a realizar el prototipo visual, los mockups de las aplicaciones web y móvil. Adjuntos en el Anexo B y C respectivamente.

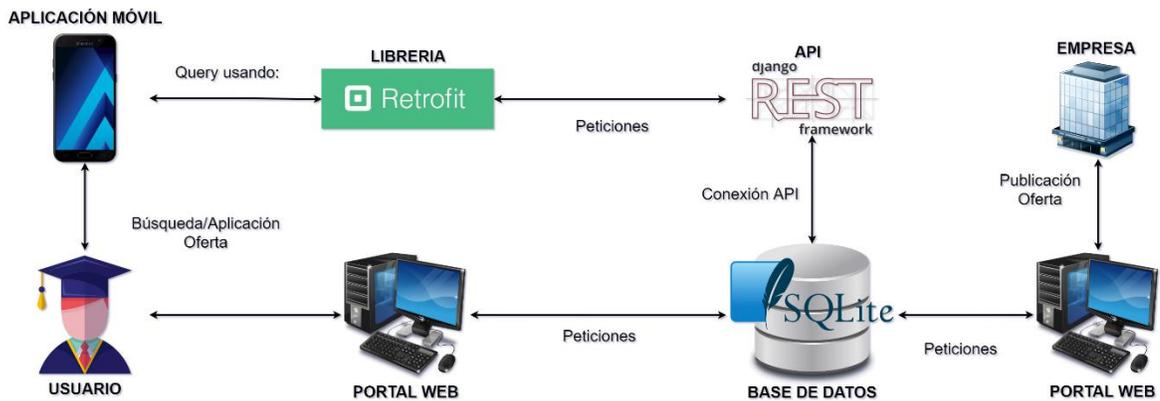
Ilustración 10: Demostración de Mockups



Fuente: Autor

- Se realiza el diagrama de arquitectura de la plataforma

Ilustración 11: Diagrama de Arquitectura



Fuente: Autor

En el diagrama se observa el funcionamiento de forma que la Empresa entra al portal web y desde su perfil publica una oferta que es guardada en la base de datos SQLite, una vez publicada la oferta, el estudiante puede visualizarla desde la aplicación web y postularse. La información se guarda igualmente en la base de datos y ahora la empresa puede ver el perfil del interesado a esa oferta específica.

En caso de la aplicación móvil, el usuario interesado accede a su cuenta desde su smartphone y la información le es mostrada gracias a la API rest framework que toda la información en formato JSON. Por medio de la librería de retrofit el usuario puede hacer solicitudes Http para postularse a la misma oferta, e igualmente la API rest framework se encarga de recibir la información y guardarla para su posterior vista.

6. Se procede a crear el proyecto en el Framework Django usando el editor de código Visual Studio Code para empezar el desarrollo del aplicativo web. Una vez creado, se crea la base de datos en SQLite por medio models.py usando lenguaje Python

Ilustración 12: Demostración de un modelo de BD

```
class oferta(models.Model):
    titulo=models.CharField(max_length=50)
    area=models.ForeignKey(area, on_delete=models.CASCADE)
    ciudad=models.ForeignKey(ciudad, on_delete=models.CASCADE)
    contenido=models.TextField()
    salario=models.PositiveIntegerField()
    horario=models.CharField(max_length=80)
    empresa=models.ForeignKey(Usuario, on_delete=models.CASCADE, blank=True)
    created=models.DateTimeField(auto_now_add=True)
    updated=models.DateTimeField(auto_now_add=True)

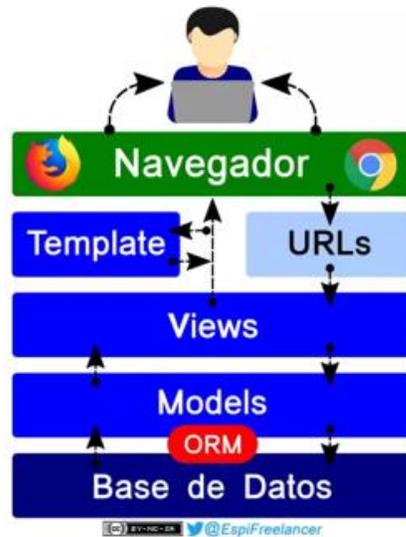
    class Meta:
        verbose_name='oferta'
        verbose_name_plural='ofertas'

    def __str__(self):
        return self.titulo
```

Fuente: Autor

Se realiza la programación de la aplicativo y todas sus funciones (views, urls y plantillas) que es la distribución como está distribuida la aplicación web para funcionar como el patrón MTV.

Ilustración 13: Jerarquía y funcionamiento MTV



```

    v aplicativopractica
      > .vscode
      > aplicativopractica
      > media
    v practiapp
      > __pycache__
      > migrations
      > static\practiapp
      > templates
      @ __init__.py
      @ admin.py
      @ apps.py
      @ filters.py
      @ forms.py
      @ models.py
      @ serializers.py
      @ tests.py
      @ urls.py
      @ views.py
      @ db.sqlite3
      @ manage.py
  
```

Fuente: Autor

En las views es la sección ocurre la reunión de los modelos de la base de datos, las plantillas ya existentes y se programan las funciones

Ilustración 14: Demostración de una View

```

class Buscar(ListView):
    model = oferta
    template_name = 'practiapp/buscar.html'

    def get(self, request):
        varofertas = oferta.objects.all()

        myfilter = filtroferta(request.GET, queryset=varofertas)
        varofertas = myfilter.qs

        data = {'ofertas':varofertas, 'myfilter':myfilter}
        return render(request, self.template_name, data)
  
```

Fuente: Autor

Ilustración 15: Demostración de la plantilla



Fuente: Autor

- Una vez realizado el aplicativo web, se desarrolla la API que usaremos para la conexión de la base de datos SQLite con el aplicativo móvil. Usando el rest framework que ofrece Django creamos la distribución para la API y sus funciones necesarias para recibir y enviar llamadas con la información correspondiente.

En las aplicaciones instaladas de Django se añade el rest_framework, para el funcionamiento de la API una clase serializers debe ser creada que especifica los datos que va a permitir mostrar esa ruta.

Ilustración 16: Implementación rest framework

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'rest_framework',  
    'practiapp',  
    'django_filters',  
    'crispy_forms',  
    'pwa',  
    'rest_framework.authtoken',  
    'django_extensions',  
]
```

Fuente: Autor

Teniendo el serializer se crea una view en el panel de views, que permite cargar los datos que se quieren mostrar a través de ese path en la API.

Ilustración 17: Demostración del serializer y el view del API

```
class ofertaSerial(serializers.ModelSerializer):  
  
    class Meta:  
        model=oferta  
        fields = ['id','titulo', 'contenido', 'salario', 'horario', 'logoempresa', 'nombreak', 'nombreak', 'nombreak', 'nombreak']
```

```
class ofertaAPI(viewsets.ModelViewSet):  
    queryset=oferta.objects.all()  
    serializer_class=ofertaSerial  
    filterset_class = OfertaFilter  
    authentication_classes = (TokenAuthentication,)  
    permission_classes = (IsAuthenticated,)
```

Django REST framework

Oferta Api List

GET /api/oferta/

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

```
{  
  "id": 1,  
  "titulo": "Ingeniero de Sistemas",  
  "contenido": "Saber python 3 años de experiencia",  
  "salario": 2500000,  
  "horario": "40 horas de lunes a viernes de libre distribución",  
  "logoempresa": "/media/perfil/birb_SgVQt5.jpg",  
  "nombreak": "admin",  
  "nombreak": "INGENIERIA",  
  "nombreak": "BUCARANGA"  
},  
{  
  "id": 2,  
  "titulo": "Medico Practicante",  
  "contenido": "Practicante del area de medicina para colaborar y prestar servicios a médico cirujano especialistas",  
  "salario": 1200000,  
  "horario": "Lunes a Sabado Horario de Madrugada",  
  "logoempresa": "/media/perfil/birb_SgVQt5.jpg",  
  "nombreak": "admin",  
  "nombreak": "MEDICINA",  
  "nombreak": "BOGOTA"  
},  
{  
  "id": 4,  
  "titulo": "Marketing Digital",  
  "contenido": "Recien graduado",  
  "salario": 1100000,  
  "horario": "Todos los dias Virtual",  
  "logoempresa": "/media/perfil/birb_SgVQt5.jpg",  
  "nombreak": "admin",  
  "nombreak": "MARKETING",  
  "nombreak": "MEDELLIN"  
}
```

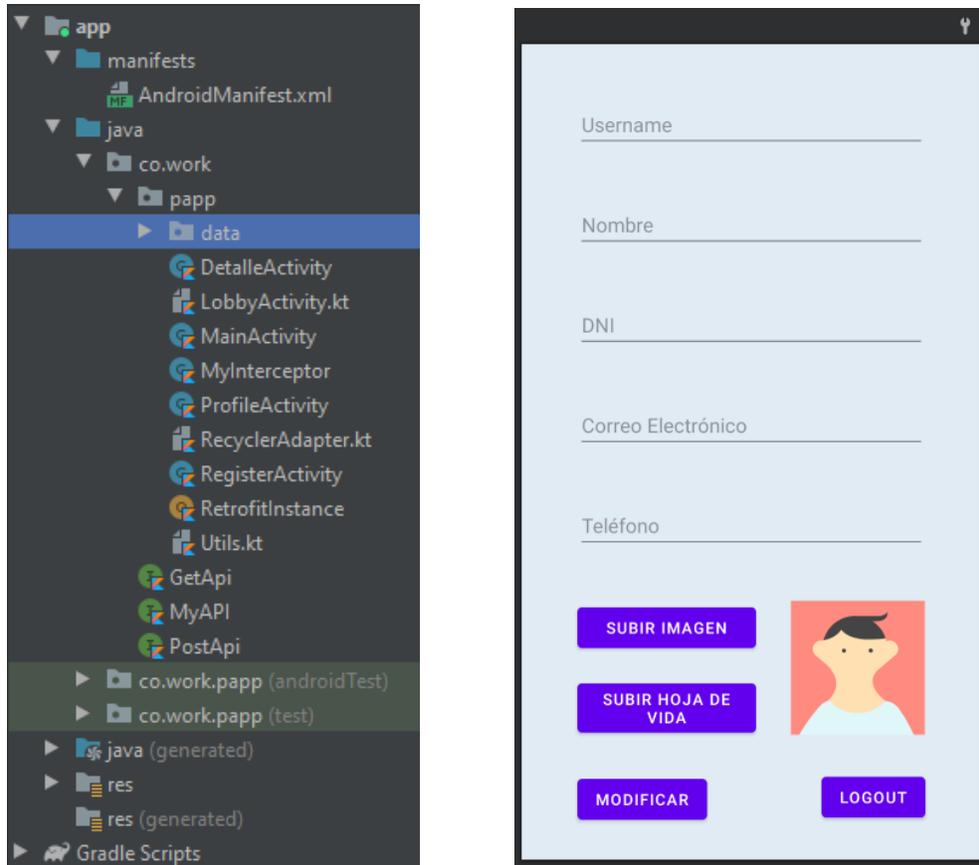
Fuente: Autor

Los PATHs principales disponibles en el api corresponden a:

- /api/usuarios: muestra todos los usuarios registrados
- /api/oferta: muestra todas las ofertas publicadas
- /api/aplicantes: muestra los interesados en cada oferta

- Para el desarrollo de la aplicación móvil se procede a crear un nuevo proyecto en Android Studio bajo el lenguaje de programación Kotlin, primeramente, se realizaron todos los layouts y distribución gráfica de todas las ventanas que tendrá disponible la aplicación móvil.

Ilustración 18: Jerarquía y un layout



Fuente: Autor

- Una vez terminado todo el Front-End de la aplicación móvil se procede a realizar el Back-End y enlazarlo mediante la librería de Retrofit, la cual permite hacer llamadas Http a un dominio solicitando o guardando datos, en este caso sería a la API realizada previamente.

Ilustración 19: Implementación de una llamada usando Retrofit

```
implementation 'com.squareup.retrofit2:retrofit:2.7.1'  
implementation 'com.squareup.retrofit2:converter-gson:2.6.1'  
implementation 'com.squareup.okhttp3:okhttp:4.9.0'  
implementation 'com.squareup.okhttp3:logging-interceptor:4.5.0'
```

```
class UserData(  
    var username: String, var nombre:String, var dni:String, var email:String, var password:String,  
    var contacto: String?): Serializable
```

```
@Headers( ...value: "Content-Type: application/json")  
@PUT( value: "api/usuarios/{id}/")  
fun update(@Path( value: "id") id: Int,  
           @Body Data3: UserData  
): Call<Map<String, Any>>
```

```
editBut.setOnClickListener { it: View!  
  
    var usernamedata = username.text.toString().trim()  
    var namedata = name.text.toString().trim()  
    var dnidata = dni.text.toString().trim()  
    var emaildata = email.text.toString().trim()  
    var contactodata = contact.text.toString().trim()  
  
    uploadFile()  
    uploadImage()  
  
    var actualdata: UserData = UserData(usernamedata!!, namedata!!, dnidata!!,  
                                       emaildata!!, password, contactodata)  
  
    retrofitBuilder.update(id, actualdata).enqueue(  
        object : Callback<Map<String, Any>> {  
            override fun onResponse(  
                call: Call<Map<String, Any>>,  
                response: Response<Map<String, Any>>  
            ) {  
                var modification = response.body()  
  
                Log.d( tag: "modificacion", modification.toString())  
            }  
  
            override fun onFailure(call: Call<Map<String, Any>>, t: Throwable) {  
            }  
        }  
    )  
}
```

Fuente: Autor

10. Después de conectar la API con nuestra aplicación móvil a través de Retrofit, programamos los botones encargados de seleccionar del móvil imagen de perfil para subir al servidor por medio de buffers finalizando la aplicación móvil y por ende la plataforma completamente.

Ilustración 20: Demostración de UploadFile

```

@Headers( ...value: "Content-Type: application/json")
@Multipart
@PUT( value: "api/usuarios/{id}/")
fun update2(
    @Path( value: "id") id: Int,
    @Part imagen: Part,
    @Body Data3: UserData
): Call<Map<String, Any>>

val body = UploadRequestBody(file, contentType: "image", callback: this)
MyAPI().uploadImage(
    id,
    MultipartBody.Part.createFormData( name: "imagen", file.name, body),
    file.asRequestBody("image/*".toMediaTypeOrNull())
).enqueue(object: Callback<Any>{
    override fun onResponse(call: Call<Any>, response: Response<Any>) {
        Log.d( tag: "subidaimagen", response.toString())
        val toast = Toast.makeText(applicationContext, text: "Imagen subida", Toast.LENGTH_LONG)
        toast.show()
    }
    override fun onFailure(call: Call<Any>, t: Throwable) {
    }
})

```

Fuente: Autor

11. Para finalizar se realizó una documentación con las pruebas de experiencia de usuario realizadas en los aplicativos. Ver Anexo D para conocer el proceso de las pruebas de experiencia de usuario en detalle.

Tabla 4: Resumen de Comparación en Pruebas

Aplicativo	Cierres Inesperados	Velocidad	Comodidad y Facilidad
Aplicación Web	No encontrados	Inmediata	Normal
API rest framework	No encontrados	Inmediata	No aplica
Aplicación Móvil	Problemas solucionados	Rápida	Sencillo

Fuente: Autor

RESULTADOS

- Documento basado en cuadros comparativos que investiga aplicaciones similares al proyecto. Este resultado se compone del Estado del Arte y el Anexo A.
- Aplicativo web funcional conectado a una base de datos activa que dispone de una REST api para responder peticiones.
- Prototipo de aplicativo móvil para Android conectado a la base de datos de la plataforma web por medio del REST api.
- Documento con resultados de la funcionalidad y calidad de la experiencia de usuario en la plataforma. Este resultado puede verificarse en el Anexo D.

CONCLUSIONES

1. El prototipo de aplicación móvil desarrollada está enfocada a facilitar el acceso al contenido de las prácticas disponibles de manera más rápida, agradable y simple para la postulación a la oferta.
2. Durante el desarrollo e investigación del proyecto se definió que el interés por parte de las empresas frente al tema de dar opciones para sus practicantes es neutral en base a la investigación de (trabajando.es, 2014), mantienen conexiones con las universidades o medios mínimos para darse a conocer.
3. En el desarrollo del prototipo de aplicación móvil se obtuvieron conocimientos en el manejo de las tecnologías relacionadas a Android Studio. Se aprendieron conceptos importantes para el desarrollo de software teniendo en cuenta las posibilidades de un dispositivo móvil. El manejo de librerías como Retrofit para las llamadas de la aplicación y conexiones con las APIs. También se obtuvieron conocimientos avanzados en el uso del lenguaje de programación Kotlin y un buen manejo de las diferentes opciones que brindan la programación a dispositivos móviles.
4. En el desarrollo de la aplicación web se obtuvieron conocimientos en el manejo de las tecnologías relacionadas a Django y SQLite. Se aprendieron conceptos importantes para el desarrollo de software web usando la herramienta mencionada El manejo de los rest framework API's para la expansión de la plataforma. También se obtuvieron conocimientos avanzados en el uso del lenguaje de programación Python y un buen manejo de las herramientas que brinda el framework Django como sus APIs que hacen posible desarrollar aplicaciones bajo la plataforma Android.
5. Durante el transcurso del desarrollo de la aplicación se implementaron funciones o pestañas adicionales respecto al modelo original, en el trayecto se detectaron falencias y mejoras que las reemplazaron haciendo más fácil la navegación de la aplicación.
6. La temática de software soportado en tecnologías móviles de este proyecto representa una gran utilidad ya que podría ser importante en otras áreas con objetivos similares, como por ejemplo en la medicina para reunir especialistas con interesados en sus servicios.

RECOMENDACIONES O TRABAJOS FUTUROS

A quien le interese para la continuidad de este proyecto las siguientes recomendaciones son métodos para expandir más o añadir funcionalidades al proyecto que tendrían un impacto positivo pero que no están incluidas dentro del modelo original:

Implementaciones:

1. Habilitar detalle en llenar formato de hoja de vida por parte de la plataforma.
2. Uso de credenciales para intentar conocer intereses del cliente y recomendar ofertas relacionadas.
3. Señalización que informe desde el menú de búsqueda al usuario estudiante en que ofertas ya ha aplicado.
4. Implementación de un chat interactivo, con la posibilidad directa de que la empresa empezando la conversación pueda en tiempo real conversar con el interesado a la postulación.
5. Implementar mejora en el sistema de listas en búsqueda de la empresa para que identifique aplicaciones previas e ilumine la casilla de la oferta ya aplicada.
6. Completar funcionalidad para subir hoja de vida desde la aplicación móvil.
7. Optimización visual general en la aplicación respecto al aspecto estético.
8. Optimización en clase de Interceptor para validación de los tokens de sesión.

REFERENCIAS BIBLIOGRÁFICAS

(s.f.). Obtenido de First Job: <https://firstjob.me/>

Android Studio. (s.f.). Obtenido de <https://developer.android.com/studio/intro/index.html?hl=es-419>

Beal, V. (s.f.). *Mobile Operating Systems (Mobile OS) Explained*. Obtenido de webopedia: https://www.webopedia.com/DidYouKnow/Hardware_Software/mobile-operating-systems-mobile-os-explained.html

Computrabajo. (s.f.). Obtenido de Play Store: <https://play.google.com/store/apps/details?id=com.redarbor.computrabajo&hl=es>

Diaz, I. A. (12 de Abril de 2014). *Que es frontend y backend en la programacion web*. Obtenido de Ser Programador: <https://serprogramador.es/que-es-frontend-y-backend-en-la-programacion-web/>

Dinero. (4 de Octubre de 2016). *El 40,8% de los jóvenes colombianos dice no conseguir trabajo por falta de experiencia*. Obtenido de Dinero: <http://www.dinero.com/economia/articulo/las-expectativas-y-condiciones-laborales-de-los-jovenes-en-colombia/222244>

Dream Jobs. (s.f.). Obtenido de Play Store: <https://play.google.com/store/apps/details?id=com.gongoplay.dreamjobs&hl=es>

Dulanga, C. (26 de Octubre de 2020). <https://medium.com/>. Obtenido de <https://medium.com/better-programming/what-will-be-the-best-backend-development-framework-for-2021-489652dee555>

EspiFreelancer. (s.f.). *EspiFreelancer*. Obtenido de <https://espifreelancer.com/mtv-django.html>

Gasca Mantilla, M. C. (s.f.). *Metodología para el desarrollo de aplicaciones móviles*. Obtenido de <http://revistas.udistrital.edu.co/ojs/index.php/Tecnura/article/view/6972/8646>

Guerrero, N. (21 de Junio de 2017). *MVC en php*. Obtenido de programaenlinea: <http://programaenlinea.net/modelo-vista-controlador-en-php/>

Job Today S.A. (s.f.). Obtenido de Play Store: <https://play.google.com/store/apps/details?id=com.jobtoday.app&hl=es>

Jobandtalent. (s.f.). Obtenido de Play Store: <https://play.google.com/store/apps/details?id=com.jobandtalent.android&hl=es>

Laboratory, L. I. (Junio de 1979). *IEEE Explore*. Obtenido de Database Machines: An Introduction: <http://ieeexplore.ieee.org.aure.unab.edu.co/document/1675377/>

- LinkedIn. (s.f.). Obtenido de Play Store:
<https://play.google.com/store/apps/details?id=com.linkedin.android.jobs.jobseeker&hl=es>
- MDN contributors. (16 de Julio de 2020). *MDN Web Docs*. Obtenido de
<https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introducci%C3%B3n>
- Mills, M. &. (2016). *Empowering Learners With Mobile Open-Access Learning Initiatives*. Kenya: Google Books. Obtenido de
<https://books.google.com.co/books?id=qpvqDQAAQBAJ&dq=defining+mobile+apps&lr=>
- Municipalidad Papudo*. (2002). Obtenido de
http://www.municipalidadpapudo.cl/reglamento_practicas_liceo.pdf
- Naeem, T. (4 de Febrero de 2021). *Astera*. Obtenido de
<https://www.astera.com/es/tipo/blog/definici%C3%B3n-de-la-API-de-descanso/>
- Pressman, R. S. (2010). *Ingeniería del Software, un enfoque práctico*. McGraw-Hill. Obtenido de <http://cotana.informatica.edu.bo/downloads/Id-Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF>
- Primerempleo. (s.f.). Obtenido de Play Store:
<https://play.google.com/store/apps/details?id=com.jobandtalent.android&hl=es>
- Rouse, M. (Julio de 2011). *Web App*. Obtenido de Search Software Quality:
<http://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app>
- Rouse, M. (20 de Febrero de 2017). *database (DB)*. Obtenido de SearchSqlServer:
<http://searchsqlserver.techtarget.com/definition/database>
- SQLite. (s.f.). *SQLite*. Obtenido de <https://www.sqlite.org/index.html>
- SynergyIndia. (s.f.). Obtenido de Play Store:
<https://play.google.com/store/apps/details?id=com.firstjob.com&hl=es>
- trabajando.es. (22 de Mayo de 2014). *El 75% de jóvenes iberoamericanos recurre primero a portales de empleo para buscar trabajo*. Obtenido de [trabajando.es: https://www.trabajando.es/contenido/noticia/3260/EI-75--de-jovenes-iberoamericanos-recurre-primero-a-portales-de-empleo-para-buscar-trabajo.html](https://www.trabajando.es/contenido/noticia/3260/EI-75--de-jovenes-iberoamericanos-recurre-primero-a-portales-de-empleo-para-buscar-trabajo.html)

ANEXOS

Anexo A: Documento basado en cuadros comparativos que investiga aplicaciones similares al proyecto

Comparación de las aplicaciones web y móvil investigadas

Se realizó la siguiente tabla que plantea los problemas y faltas que se pudo observar en las aplicaciones similares investigadas en la tabla anterior, junto al problema encontrado se ha descrito una breve solución por parte de este proyecto para no cometer el mismo error.

Apps	Problema	Solución del proyecto
Firstjob.me	No tiene App Móvil	Se realizará un prototipo de aplicación móvil
First Job	Únicamente disponible en inglés	Será desarrollada en español, dependiendo de su lanzamiento y éxito se consideraría su traducción para un nuevo público
JobToday, First Job, Primerempleo	Disponible únicamente para algunas zonas o países	Planeada para ser accesible a todas las empresas en todo lugar que deseen publicar
Jobandtalent, Dream Jobs	Falta de empresas asociadas al servicio en varias zonas donde lo prestan	Planeada para ser accesible y darse a conocer a todas las empresas
Computrabajo, LinkedIn	Las ofertas publicadas suelen pedir más requisitos de los necesarios al ser también servicios planeados para todo tipo de trabajo al público	Al únicamente enfocarnos en prácticas y primer empleo, solicitar requisitos aptos y lógicos para este tipo de ofertas por parte de las empresas
Computrabajo, LinkedIn	Pocas ofertas de categoría prácticas disponibles y primer empleo	Al únicamente enfocarnos en prácticas y primer empleo, aumentar en grandes medidas las ofertas disponibles para cada zona

Estudio comparativo de Frameworks de desarrollo web

En la siguiente tabla se muestra una pequeña investigación realizada acerca de back-end Frameworks para la realización de un aplicativo web con el que se busca poder compararlos y seleccionar uno para el proyecto.

En la siguiente tabla, los Frameworks seleccionados anteriormente fueron comparados desde un punto de vista más técnico y estadístico.

Framework	Descripción	Lenguaje
NodeJS	NodeJS es un JavaScript Runtime que se puede utilizar para fines de desarrollo multiplataforma. Dado que JavaScript es uno de los lenguajes más populares en el contexto actual, esa popularidad ha elevado a NodeJS a ser uno de los Frameworks de back-end más utilizados.	JavaScript
Django	Django es un marco de aplicación web de código abierto y de alto nivel escrito en Python. Django se introdujo en 2005 y su idea de usar Python para el desarrollo web fue una gran revolución. Django sigue la arquitectura modelo-plantilla-vista (MTV) y el enfoque principal de este marco es proporcionar un método sencillo para el desarrollo de sitios web complejos.	Python
Laravel	PHP es otro lenguaje famoso entre los desarrolladores web y Laravel está basado en PHP. Laravel sigue la arquitectura modelo-vista-control y es robusto y fácil de entender. Proporciona un gran conjunto de funciones, como enrutamiento flexible para escalar fácilmente, gestión de la configuración para manejar diferentes entornos, constructores de consultas y ORM para consultar bases de datos.	PHP

¿Quién lo usa?	
NodeJS	Netflix, Paypal, Uber, LinkedIn, Ebay
Django	Instagram, The New York Times, Pinterest, Nasa Science, National Geographic, Mozilla
Laravel	Deltanet Travel, Neighborhood Lender. MyRank, 9Gag, MasterCard

	NodeJS	Django	Laravel
Programming Language	JavaScript	Python	PHP
GitHub Stars	73.8k	53k	62.1k
Community Size	Large	Large	Large
Learning Curve	Easy to learn but can be challenging if you are new to JavaScript	Easy to learn since Python is straight forward.	Basic knowledge in PHP will do the trick.
Scalability	Mid-Range	High	Mid-Range
Performance	High	Mid-Range	Mid-Range
Security	Medium	High	Medium
Rapid Development	Recommended	Most Recommended	Not Recommended

Adaptado de “NodeJS vs. Django vs. Laravel - What Will Be the Best Backend Development Framework for 2021”, de Chameera D., (10, 26, 2020). Recuperado de <https://medium.com/better-programming/what-will-be-the-best-backend-development-framework-for-2021-489652dee555>

Selección web

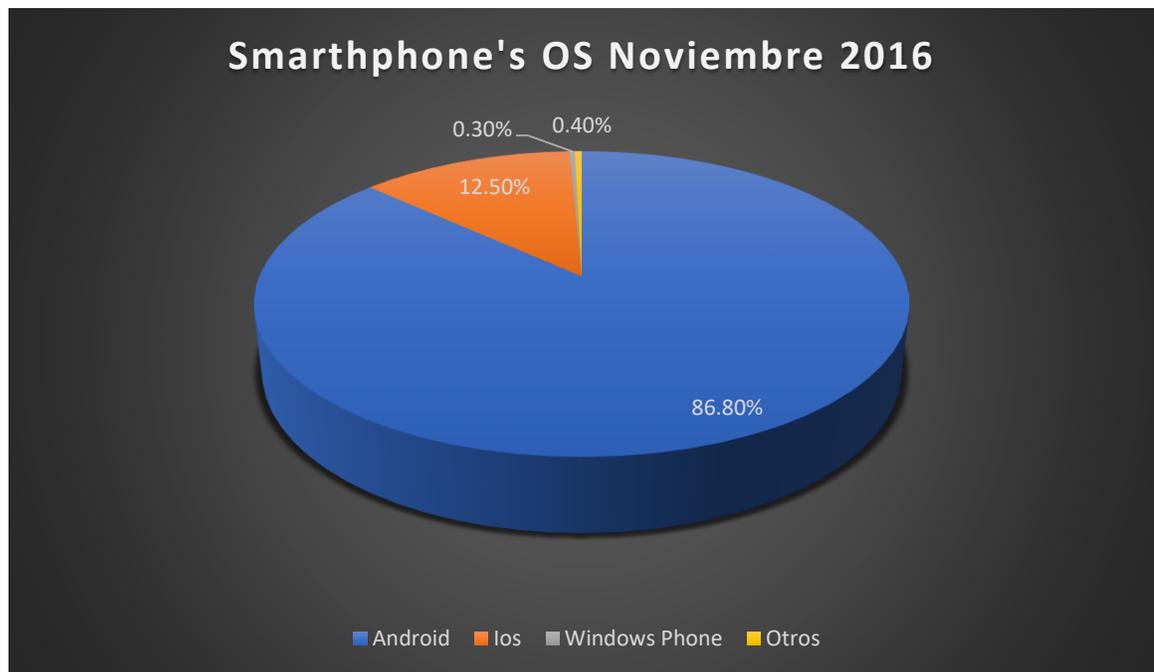
El framework web seleccionado fue Django al tener un soporte total por parte del lenguaje de alto nivel Python, esto comprueba que es muy estable y seguro para ser programado en todo sentido.

Entre los Frameworks de Python se ha seleccionado a Django por encima de Flask por ser más estructurado y seguro a pesar que Flask sea más fácil de aprender.

Para el front-end se han elegido las vistas de Django, por tener fácil acceso y manejo de estas directamente desde el mismo framework demostrando que es muy completo.

Selección Móvil

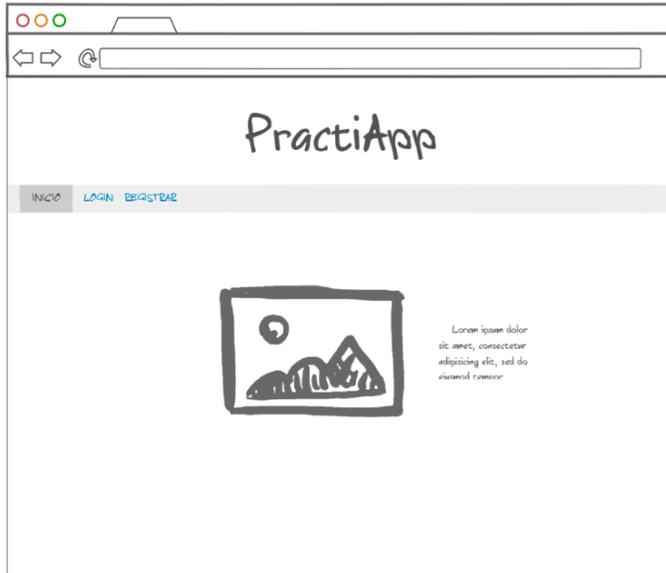
La siguiente ilustración evidencia casi en la actualidad el enfoque de los sistemas operativos de smartphones del Mercado por lo que se ha decidido realizar la aplicación para el Sistema operativo Android por la dominancia en el Mercado.



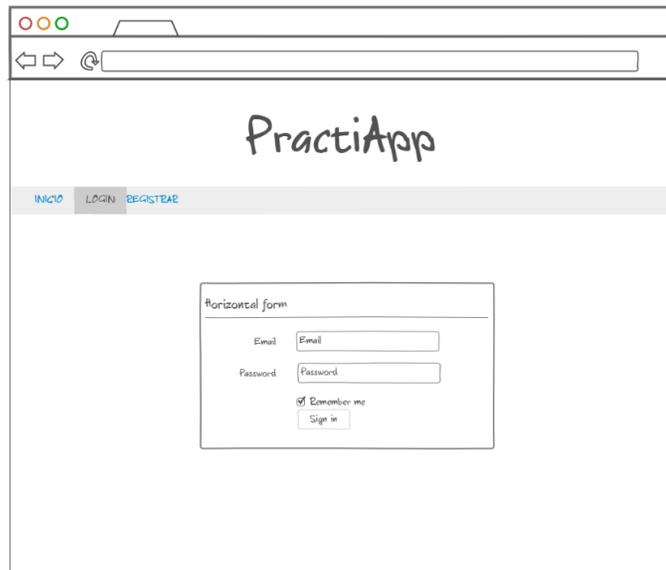
Recuperado de https://www.webopedia.com/DidYouKnow/Hardware_Software/mobile-operating-systems-mobile-os-explained.html

Anexo B: Mockups del Aplicativo Web

Pantalla de inicio



Login



Registro

A browser window showing the registration page of PractiApp. The page has a navigation bar with 'INICIO', 'LOGIN', and 'REGISTRAR'. The main heading is 'Registro de usuario'. The form includes a 'empresa?' checkbox, a 'checkbox' checkbox, and several text input fields for 'username', 'nombre', 'nit', 'email', 'contraseña', and 'confirmar contraseña'. A 'Registrar' button is at the bottom.

PractiApp

INICIO LOGIN REGISTRAR

Registro de usuario

empresa? checkbox

username

nombre

nit

email

contraseña

confirmar contraseña

Registrar

Búsqueda de Ofertas

A browser window showing the search for offers page of PractiApp. The page has a navigation bar with 'INICIO', 'OFERTAS PUBLICADAS', 'PUBLICAR OFERTA', and 'PERFIL'. The main heading is 'PractiApp'. The form includes an 'Inline form' with 'Email', 'Password', 'Remember me', and 'Sign in' fields. Below the form is a table with three columns: 'header 1', 'header 2', and 'header 3'. Each row contains 'text' and a 'Ver' button.

PractiApp

INICIO OFERTAS PUBLICADAS PUBLICAR OFERTA PERFIL

Inline form

Email Password Remember me

header 1	header 2	header 3	
text	text	text	<input type="button" value="Ver"/>
text	text	text	<input type="button" value="Ver"/>
text	text	text	<input type="button" value="Ver"/>

Oferta en Detalle

PractiApp

INICIO OFERTAS PUBLICADAS PUBLICAR OFERTA PERFIL

Oferta #x

Título Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non

Area

Ciudad

Descripción

Salario

Horario

Empleado

Aplicar

Publicación Oferta

PractiApp

INICIO OFERTAS PUBLICADAS PUBLICAR OFERTA PERFIL

Registro de Oferta

Título
Texte input

Ciudad
Select

Area
Select

Salario
Texte input

Horario
Texte input

Descripción
Texte area

Fuente: Autor

Anexo C: Mockups del Aplicativo Móvil

Login



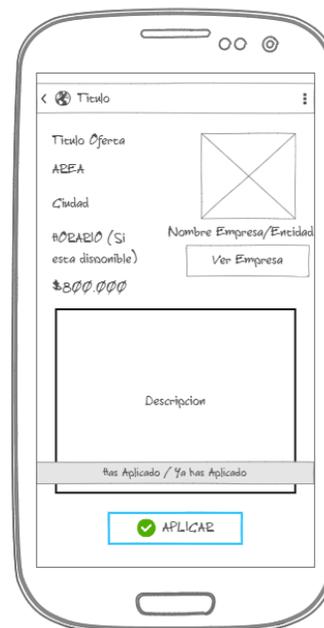
Registro/Perfil



Búsqueda



Oferta en Detalle



Fuente: Autor

Anexo D: Documentación de pruebas de experiencia de usuario

Se definió que las únicas pruebas que se realizarían a la plataforma serían las de experiencia de usuario.

Se realizaron pruebas de experiencia de usuario a toda la plataforma, ambos aplicativos, web y móvil para determinar varios factores como comodidad, funcionamiento y facilidad. Las pruebas se realizaron montando la plataforma en una dirección local.

La aplicación web fue la primera en ser probada al ser la principal, se tomo por simulacro de ambos tipos de registro, usuario y empresa, durante las pruebas en la cual ambos registraban sus cuentas en el sistema, editaban su perfil con la información necesaria y por parte de las empresas creaban una oferta para que el usuario pudiera aplicar. El usuario podía aplicar a la oferta y luego este aparecería en el menú de aplicantes de la oferta creada por aquella empresa. Inconvenientes encontrados dentro de esta prueba fueron que la imagen de perfil no se podía visualizar correctamente y el usuario no podía saber a cuáles ofertas ya había aplicado, por lo que se creó otra interfaz en la cual podía ver su historial de aplicaciones.



Fuente: Autor

La aplicación móvil fue probada usando el emulador de Android Studio, hubo inconvenientes la primera vez, después de arreglos en la programación de Retrofit se realizó la prueba de experiencia de usuario, todas las funcionalidades cumplieron con su cometido, pero había ciertas funciones que eran poco cómodas en móviles

por lo que se modificaron estas vistas, dejando la búsqueda avanzada en el mismo menú de la lista y eliminando la vista dedicada que tenía.



Fuente: Autor

En general la experiencia de la aplicación web fue mucho más fluida que la aplicación móvil, sin embargo, ambas demostraron tener rápidos tiempos de respuesta y disponer de una interfaz bastante intuitiva y sencilla para ser usada por cualquier usuario nuevo.

Las pruebas se hicieron creando varios perfiles de usuarios y empresas, en la cual distintas empresas crearon 10 ofertas nuevas y cada usuario aplico a 5 de ellas, para en el final cada cuenta de empresa revisara que cada perfil del aplicante era debido y correcto, a continuación, se explica el contenido de cada prueba en detalle en base a sus funcionalidades.

Funcionalidades Aplicativo Web:

- Permite logear usuarios y empresas
- Editar perfil usuarios y empresas
- Publicar ofertas desde empresa
- Revisar aplicantes desde empresa
- Buscar y aplicar a ofertas desde usuario
- Visualizar ofertas ya aplicadas desde usuario
- Encontrar ofertas con filtros de búsqueda desde usuario

Pruebas de Experiencia de Usuario de la aplicación Web

Prueba: Registro de Cuenta y Login

Descripción: Comprobar que los usuarios pueden crear una cuenta registrándose y logearse correctamente a ella.

Resultados Esperados: Usuarios crean cuenta y acceden correctamente

Resultados Obtenidos: Usuarios crearon cuenta y accedieron correctamente

Prueba: Modificación de Perfil

Descripción: Comprobar que los usuarios modificar los datos de su perfil y reemplazarlos por otros correctamente.

Resultados Esperados: Usuarios modifican información de perfil satisfactoriamente

Resultados Obtenidos: Usuarios modificaron su perfil sin inconvenientes

Prueba: Publicación de Oferta

Descripción: Comprobar que las empresas pueden crear ofertas con todos los filtros disponibles

Resultados Esperados: Empresas pueden publicar ofertas usando las herramientas de la página web

Resultados Obtenidos: Empresas logran crear ofertas satisfactoriamente

Prueba: Modificación de Oferta

Descripción: Comprobar que las empresas pueden modificar, borrar las ofertas antes publicadas

Resultados Esperados: Empresas pueden modificar o eliminar ofertas satisfactoriamente

Resultados Obtenidos: Empresas logran borrar o modificar ofertas antiguas sin problemas

Prueba: Ver aplicantes de Ofertas

Descripción: Comprobar que las empresas pueden ver los perfiles y hojas de vida de los usuarios que han aplicado a cada una de sus ofertas respectivamente.

Resultados Esperados: Empresas pueden ver la información de los aplicantes

Resultados Obtenidos: Empresas logran ver la información de los aplicantes correctamente

Prueba: Aplicación a Ofertas

Descripción: Comprobar que los usuarios pueden aplicar a cualquier oferta de su interés y la información se guarda correctamente

Resultados Esperados: Usuarios aplican satisfactoriamente a ofertas

Resultados Obtenidos: Usuarios logran aplicar satisfactoriamente a ofertas.

Prueba: Búsqueda de Ofertas

Descripción: Comprobar que los usuarios pueden buscar todas las ofertas publicadas y haciendo uso de los filtros de búsqueda.

Resultados Esperados: Usuarios pueden buscar ofertas

Resultados Obtenidos: Usuarios logran buscar ofertas satisfactoriamente

Prueba: Ver listado de ofertas aplicadas

Descripción: Comprobar que los usuarios tienen acceso a una tabla que lista todas las ofertas que han decidido aplicar

Resultados Esperados: Los usuarios pueden ver el historial de sus aplicaciones

Resultados Obtenidos: Los usuarios logran ver todas sus aplicaciones realizadas listadas

Funcionalidades Aplicativo Móvil:

- Permite logear usuarios
- Editar perfil usuarios
- Buscar y aplicar a ofertas desde usuario
- Encontrar ofertas con filtros de búsqueda desde usuario

Pruebas de Experiencia de Usuario de la aplicación Móvil

Prueba: Registro de Cuenta y Login

Descripción: Comprobar que los usuarios pueden crear una cuenta registrándose y logearse correctamente a ella.

Resultados Esperados: Usuarios crean cuenta y acceden correctamente

Resultados Obtenidos: Usuarios crean cuenta y acceden correctamente

Problemas Inesperados: Los usuarios deben conectarse cada vez que entran a la aplicación

Solución: Se guarda la sesión activa con el uso de sharedpreferences para guardar el id guardado previamente y se añade botón de Logout en Perfil.

Prueba: Modificación de Perfil

Descripción: Comprobar que los usuarios modificar los datos de su perfil y reemplazarlos por otros correctamente.

Resultados Esperados: Usuarios modifican información de perfil satisfactoriamente

Resultados Obtenidos: No todos los elementos cambian en la base de datos y el archivo PDF o de imagen no se sube al servidor correctamente.

Solución: Se adaptó la llamada de Retrofit para subir los archivos por buffers junto a la información que desee cambiarse en los respectivos cambios correctamente.

Prueba: Búsqueda de Ofertas

Descripción: Comprobar que los usuarios pueden buscar todas las ofertas publicadas y haciendo uso de los filtros de búsqueda.

Resultados Esperados: Usuarios pueden buscar ofertas

Resultados Obtenidos: Solo se puede traer datos la primera vez, las búsquedas con el layout de búsqueda avanzada no funcionan correctamente y son incómodos.

Solución: Se ha adaptado la búsqueda avanzada en el mismo layout donde aparecen las ofertas, para que no sea necesario cambiar de ventana en cada proceso y ahora cada búsqueda funciona correctamente.

Prueba: Aplicación a Ofertas

Descripción: Comprobar que los usuarios pueden aplicar a cualquier oferta de su interés y la información se guarda correctamente

Resultados Esperados: Usuarios aplican satisfactoriamente a ofertas

Resultados Obtenidos: Usuarios logran aplicar satisfactoriamente a ofertas.

Adiciones: Se añade función para que le advierta al usuario que ya ha aplicado a esa oferta antes.

A continuación, unas tablas donde se resumen los resultados de cada una de las pruebas mencionadas anteriormente:

Aplicativo	Cierres Inesperados	Velocidad	Comodidad y Facilidad
Aplicación Web	No encontrados	Inmediata	Normal
API rest framework	No encontrados	Inmediata	No aplica
Aplicación Móvil	Problemas solucionados	Rápida	Sencillo

Aplicativo	Login/Registro	Modificación Perfil	Buscar/Aplicar Ofertas
Aplicación Web	Satisfactorio	Satisfactorio	Satisfactorio
Aplicación Móvil	Satisfactorio	No Satisfactorio	No Satisfactorio

Aplicativo	Publicar Ofertas	Ver Aplicantes	Ver Ofertas Aplicadas
Aplicación Web	Satisfactorio	Satisfactorio	Satisfactorio
Aplicación Móvil	No Aplica		

Fuente: Autor

*Los escenarios con resultados “No Satisfactorio” fueron solucionados con variados arreglos o modificaciones para poder cumplir con sus funcionalidades debidas sin dejar de ser cómodo para el usuario.