



## **Universidad de la Defensa Nacional - CRUC IUA**

**FACULTAD DE INGENIERÍA**

**"DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE  
INDICADORES RELACIONADOS A PROGRAMAS DE FORMACIÓN"**

**ANTEPROYECTO DE TRABAJO FINAL DE GRADO**

**Carrera: Ingeniería en Informática**

**Director: Pérez, Sofia**

**Autores: Blanc, Facundo - Brain, Leonardo**

## **TÍTULO DEL PROYECTO**

Diseño, desarrollo e implementación de un sistema de gestión de indicadores relacionados a programas de formación.

## **OBJETIVO DEL PROYECTO**

Objetivo general:

Contribuir, posibilitar y facilitar el seguimiento de los programas de formación e inclusión tecnológica para mujeres adolescentes mediante el desarrollo e implementación de un sistema de seguimiento, control y obtención de métricas para estos programas. A raíz de estos indicadores se busca encontrar señales y problemáticas a resolver que ayuden a disminuir la brecha de género en el área de la tecnología en la región.

Objetivos específicos:

- Proporcionar una herramienta que según los datos almacenados en ella permita obtener indicadores de:
  - Qué porcentaje de mujeres adolescentes completan los cursos y luego inician una carrera relacionada a la tecnología.
  - Qué porcentaje de mujeres adolescentes abandonaron los cursos, lugar y motivos.
  - Qué porcentaje de mujeres adolescentes viven con familiares y cómo es su parentesco.
  - Qué cursos poseen más inscriptos y cuales más desertores.
  - Situación socio-económica de las mujeres adolescentes que realizan los cursos.
  - Preferencias de carreras universitarias o terciarias de estas mujeres
  - Qué porcentaje de mujeres trabajan y estudian al mismo tiempo.
- Identificar y permitir la visualización de indicadores relevantes y representativos que ayuden al seguimiento de los programas.
- Proporcionar un sistema de notificaciones que comunique a los distintos tipos de usuarios sobre eventos que puedan ser previamente configurados por ellos.
- Permitir la configuración de distintos perfiles con diferentes niveles de privilegios para utilizar el sistema.

## **DESTINATARIOS**

El Trabajo Final de Grado se realizará en el marco del proyecto UNDEX 775/2019 aprobado al Depto. de Computación e Informática de la Facultad de Ingeniería del CRUC IUA (Resolución 441/2019), en la convocatoria 2019 de UNDEF [?]. En dicho proyecto se desarrollará la estrategia a implementar mediante un sistema software para gestionar el seguimiento de los programas de capacitación tecnológica para favorecer la inclusión de la mujer.

## **BENEFICIOS ESPERADOS**

- Obtención de indicadores y métricas para toma de decisiones.
- Estandarización, limpieza y centralización del almacenamiento de la información de los cursos evitando pérdidas de la misma.
- Automatización de la carga de datos correspondiente a los cursos, lo que conlleva a un menor esfuerzo por parte de las personas encargadas de realizar estas tareas.
- Seguimiento de las alumnas participantes y de los programas.
- Facilidad de acceso a la información.

## **ESTUDIO TÉCNICO**

- Para el desarrollo del software se utilizarán 3 lenguajes de programación diferentes:
  - Para el Backend se utilizarán los lenguajes Golang y Python.
  - Para el Frontend se utilizará Javascript apoyándose en un framework llamado React también basado en Javascript.
- Se utilizarán tanto Bases de Datos Relacionales como No Relacionales para el guardado de datos.
- Se utilizará el software RabbitMQ para lograr comunicación entre diferentes microservicios.
- Para desplegar el sistema se utilizará un servidor provisto por el CRUC IUA.

## **FORMULACIÓN Y VALORACIÓN DE ALTERNATIVAS**

No se han detectado alternativas gratuitas y/o herramientas que favorezcan y ayuden al seguimiento de los programas de inclusión.

## METODOLOGÍA

Para comenzar se procederá al estudio general de los distintos programas de formación tecnológica para la inclusión de la mujer y se hará un relevamiento de la información disponible de los cursos anteriores de este programa para poder identificar qué datos serían de interés a la hora de ingresarlos al sistema y qué información debería proveer el mismo.

Luego se seguirá con un análisis de las funcionalidades del sistema para poder hacer el diseño del mismo.

Una vez que el diseño esté finalizado, se comenzará la implementación del sistema.

Paralelamente, se desarrollarán y ejecutarán pruebas para asegurar el correcto funcionamiento del mismo.

Finalmente, se procederá a realizar pruebas del sistema completo intentando imitar lo mejor posible al funcionamiento real del mismo.

Para llevar a cabo todas estas actividades se utilizará una metodología Ágil, en este caso Scrum.

Se utilizarán iteraciones llamadas Sprints que tendrán una duración preestablecida de entre 2 y 4 semanas, logrando con esto obtener una retroalimentación continua y rápida del desarrollo del sistema.

Para la organización y planificación de las tareas se utilizará el sistema de Trello con su versión FREE.

Para realizar las actividades requeridas para la implementación del sistema se definirán tareas semanalmente apoyándose en el tablero de SCRUM que proporciona el sistema Trello. Estas tareas se definirán y dividirán entre los dos estudiantes encargados de realizar el Trabajo Final de Grado.

Durante la implementación del sistema se realizarán tres tipos de pruebas paralelamente para asegurar el correcto funcionamiento del mismo:

- Pruebas unitarias.
- Pruebas de integración.
- Pruebas funcionales.

El equipo estará formado por los siguientes roles:

- Scrum Master: Será la directora de tesis Sofia Pérez.
- Product Owner: Será la directora de tesis Sofia Pérez.

- **Sprint Team:** Estará compuesto por los dos estudiantes encargados de realizar el proyecto.

## RESUMEN TÉCNICO

### Arquitectura

Para el desarrollo de este sistema se eligió una arquitectura orientada a microservicios, es decir, se construirá la aplicación como un conjunto de pequeños servicios que se ejecutarán de manera independiente y autónoma y cada uno de ellos implementará una funcionalidad puntual. Estos servicios se comunicarán entre sí mediante diferentes mecanismos para así poder cumplir con el funcionamiento completo del sistema.

La razón por la cual se eligió esta arquitectura fue por las ventajas que esta ofrece:

- **Modularidad:** Al tratarse de servicios autónomos, se pueden desarrollar y desplegar de forma independiente. Además, un error en un servicio no debería afectar la capacidad de otros servicios para seguir trabajando según lo previsto.
- **Escalabilidad:** Como es una aplicación modular, se puede escalar horizontalmente cada parte según sea necesario, aumentando el escalado de los módulos que tengan un procesamiento más intensivo.
- **Versatilidad:** Se pueden usar diferentes tecnologías y lenguajes de programación, lo que permite adaptar cada funcionalidad a la tecnología más adecuada y rentable.
- **Rapidez de actuación:** El reducido tamaño de los microservicios permite un desarrollo menos costoso, así como el uso de “contenedores de software” permite que el despliegue de la aplicación se pueda llevar a cabo rápidamente.
- **Mantenimiento simple y barato:** Al poder hacerse mejoras de un solo módulo y no tener que intervenir en toda la estructura, el mantenimiento es más sencillo y barato que en otras arquitecturas.
- **Agilidad:** Se pueden utilizar funcionalidades típicas (autenticación, trazabilidad, etc.) que ya han sido desarrolladas por terceros, no hace falta que el desarrollador las cree de nuevo.

A continuación, se observa un gráfico de la arquitectura de microservicios propuesta.

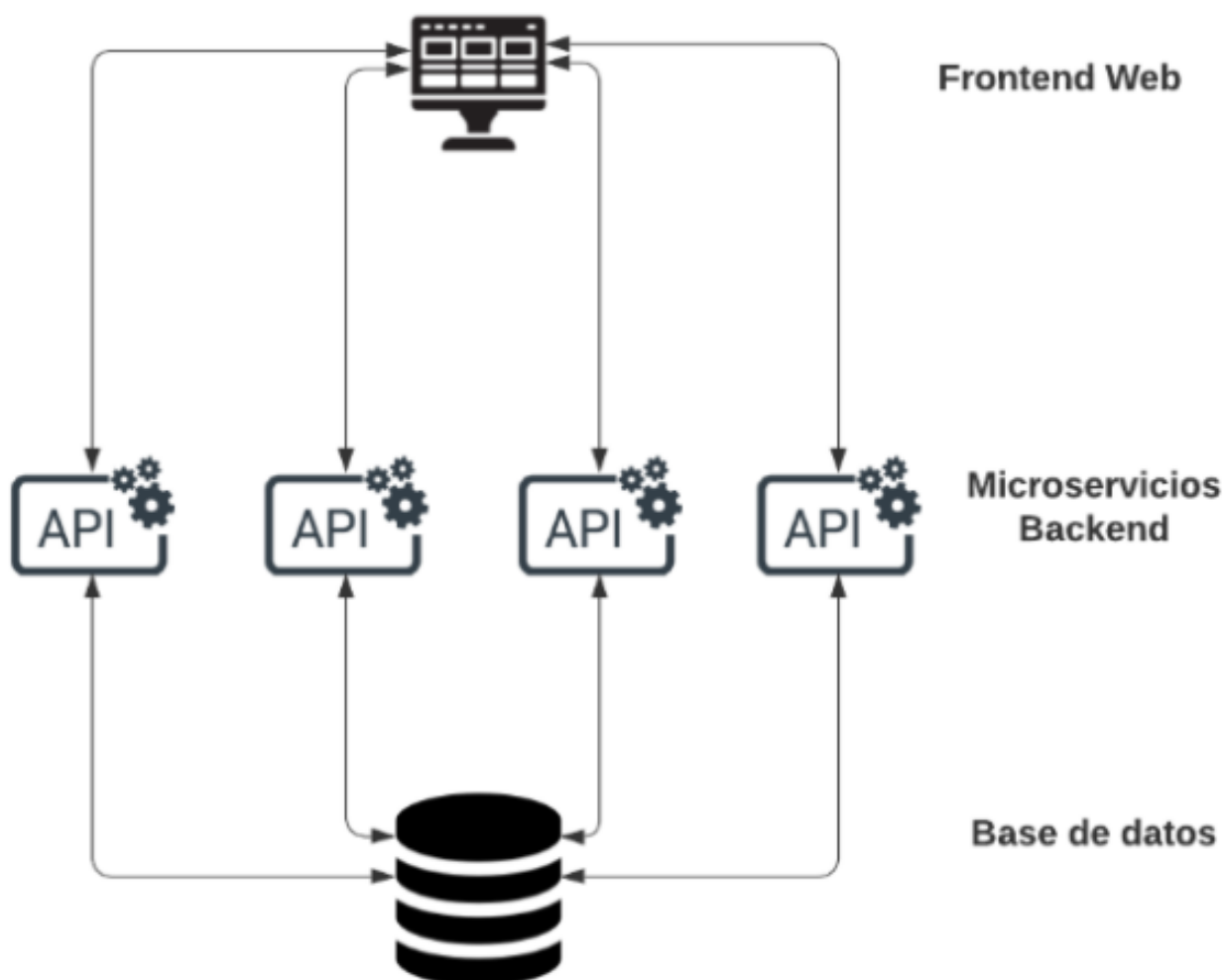


Figura 1: Diagrama de bloques

### Estrategias de despliegue

En cuanto a la forma en la que se desplegará el sistema se optó por utilizar la combinación de docker y kubernetes.

Algunas de las ventajas de utilizar contenedores de Docker son las siguientes:

- **Aislamiento:** Se envía al contenedor todo el código junto con todas las dependencias necesarias para que el servicio funcione. Esto resuelve el problema de que muchas veces las cosas funcionan en las máquinas locales y al momento de desplegarlo en un ambiente productivo no, por problemas de dependencias y/o versionado de código.
- **Portabilidad:** Como parte de este aislamiento y con la ayuda del servicio de docker se logra que la misma imagen de Docker pueda correr sin cambios en una computadora de escrito-

rio, una notebook, un datacenter, una máquina virtual o en los diferentes proveedores de servicios en la Nube.

- **Rapidez de entrega:** Se pueden construir y desplegar versiones de cada servicio de manera rápida por lo que esto ayuda a agilizar las entregas.
- **Facilidad en las pruebas:** Docker puede hacer que las pruebas de control de calidad sean más fáciles, rápidas y efectivas. Los contenedores se pueden configurar para que solo contengan una parte o partes de toda la configuración del entorno. De esta forma se puede construir pequeños subconjuntos de pruebas exploratorias y ser más efectivos para encontrar el problema. A todo esto, hay que sumarle que el contenedor puede tener exactamente la misma configuración que el entorno de producción, por lo que se evita sorpresas desagradables.

Para la orquestación de estos contenedores de Docker es cuando Kubernetes entra en juego. Esto no es más que un sistema de código abierto construido por Google que ayuda con la implementación, escalado y administración de aplicaciones en contenedores.

El esquema de contenedores bajo el que trabaja Kubernetes permite que varias partes de un mismo proyecto o aplicación se ejecuten en entornos (contenedores) aislados. De este modo, si uno de los contenedores está inoperativo por algún motivo, se puede restaurar únicamente esa parte del proyecto, y no todo el paquete o servidor. La utilización de esta tecnología brinda una serie de beneficios como escalabilidad y balanceo de carga, aislamiento de procesos y aplicaciones, facilidad de despliegue, optimización de recursos de forma automática, alta disponibilidad, entre otras. También es muy importante destacar que Kubernetes permite desplegar la aplicación tanto en un servidor físico como en los distintos proveedores en la Nube o también realizar una implementación híbrida.

Otra ventaja muy grande de utilizar Docker y Kubernetes es que se pueden integrar con herramientas de Integración y Entrega Continua y herramientas de gestión como por ejemplo aplicaciones de logueo, aplicaciones para gestionar el rendimiento del sistema, aplicaciones para gestión de errores, entre otras.

## Lenguajes de Programación

- Para el desarrollo del backend, se optó por los lenguajes Golang y Python ya que son lenguajes conocidos por los que llevaran a cabo el sistema y además son lenguajes eficientes en cuanto al procesamiento de datos.
- Para el desarrollo del frontend, se optó por utilizar Javascript con la ayuda del framework React.

## Base de Datos

Para el almacenamiento de los datos se eligió utilizar MySQL ya que es tecnología gratuita, fácil de utilizar y rápida. Es una tecnología muy flexible con todas las versiones de Linux, UNIX y Windows y por otro lado, el alto rendimiento que generan las bases de datos MySQL es muy bueno.

## Sistema de colas

Se utilizará un servicio de encolado de mensajes llamado RabbitMQ. Esto ayuda a que los servicios se comuniquen de manera asíncrona para cumplir con las diferentes funcionalidades requeridas.

## PROGRAMACIÓN DE ACTIVIDADES

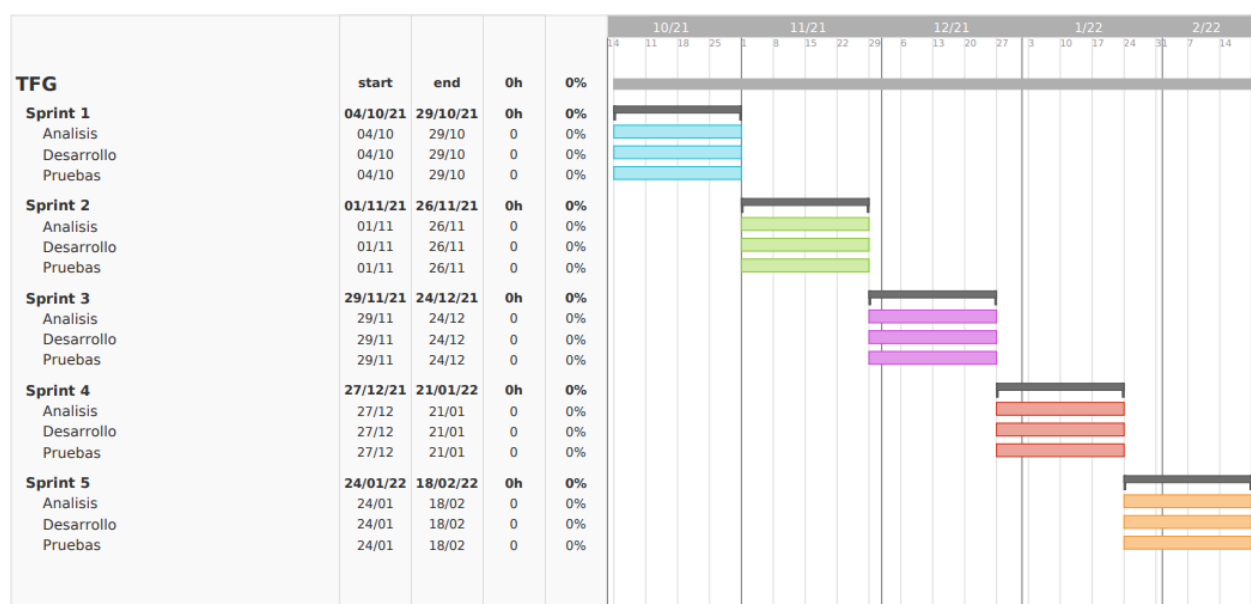


Figura 2: Diagrama de Gantt

## PROGRAMACIÓN DE RECURSOS

Se utilizarán los siguientes recursos y plataformas de desarrollo:

- Dos estaciones de trabajo con macOS Big sur, versión 11.2.3.
- Entornos de programación de JetBrains con licencia educativa.
- Visual Studio Code.



- Trello para el seguimiento de las tareas.
- Oracle VirtualBox.
- Github como versionador de código.
- Jenkins como herramienta de Integración Continua y Despliegue Continuo (CI/CD).
- Postman free edition.

El producto se desplegará sobre un servidor provisto por el Departamento de Computación e Informática, para publicarlo bajo un subdominio de su página web.

## **FACILIDADES REQUERIDAS AL CRUC IUA**

Se hará uso de los equipos e infraestructura disponible en el Departamento de Computación e Informática.

## **PRESUPUESTO**

(a) Estimación de costos de equipamiento a utilizar en el proyecto:

Valor de un servidor físico con características necesarias:

- Memoria RAM 16GB DDR4.
- Disco duro SATA de 4TB.
- Procesador escalable Intel® Xeon® 3204 (6 núcleos, 1,9 GHz, 85 W).
- Fuente de alimentación 550w.
- Estado: Disponible y provisto por el Depto. de Computación e Informática de la Facultad de Ingeniería del CRUC IUA. Costo: \$0.-.

(b) Estimación de costos de desarrollo:

- Se estima que el costo de desarrollo para realizar este proyecto es de 3600 USD, siendo el valor en hora de trabajo de 20 USD, con un total de 180 horas estimadas.

## **FUENTES DE FINANCIAMIENTO**

No corresponde por tratarse de un trabajo final para obtener el título de grado de la carrera.

## **RIESGOS ESPERADOS Y SUPUESTOS ASUMIDOS**

Los riesgos esperados son:

- Corte de luz, caída de los dominios del CRUC IUA o cualquier otro tipo de incidente que perjudique el dominio en donde estará desplegado el sistema ya que los tesisistas no se harán cargo del Hosteo.
- Si bien los tesisistas no son expertos en el dominio, sí lo es su directora de tesis.

Los supuestos asumidos son:

- Los equipos provistos por el CRUC IUA estarán disponibles desde la fecha de inicio del proyecto.
- El equipo del proyecto UNDEX estará disponible para brindar la información necesaria y requerida en el proyecto, así como realizar pruebas y toda actividad derivada de la metodología de desarrollo empleada.

## **INVERSIÓN REQUERIDA**

Para el desarrollo de este Trabajo Final de Grado no se requiere ninguna inversión.

## **PROYECCIÓN DE COSTOS DE OPERACIÓN Y MANTENIMIENTO**

El sistema se desplegará en un servidor físico por lo cual los costos de operación y mantenimiento serán los relacionados a los gastos de luz e internet que serán a cargo del CRUC IUA.

La forma en la que será desplegado el sistema en este servidor será la siguiente:

- Los microservicios que componen el sistema estarán desplegados con la ayuda de la tecnología de Docker y alguna herramienta para la gestión de estos contenedores.
- En cuanto a la base de datos, se utilizará una instalación nativa de MySQL en el servidor físico.
- En el caso de necesitar de algún otro servicio que no sea de persistencia de datos se utilizará también la estrategia de despliegue en un contenedor de Docker.
- En el caso de necesitar algún otro servicio de persistencia de datos se procederá a utilizar la misma estrategia que con la base de datos MySQL, es decir, se instalará de manera nativa en el servidor.

Como alternativa se evaluó la opción de desplegar el sistema utilizando algún proveedor en la nube. En este caso se optó por Amazon Web Services por ser uno de los más conocidos y confiables. La sugerencia de despliegue utilizando Amazon Web Services es la siguiente:

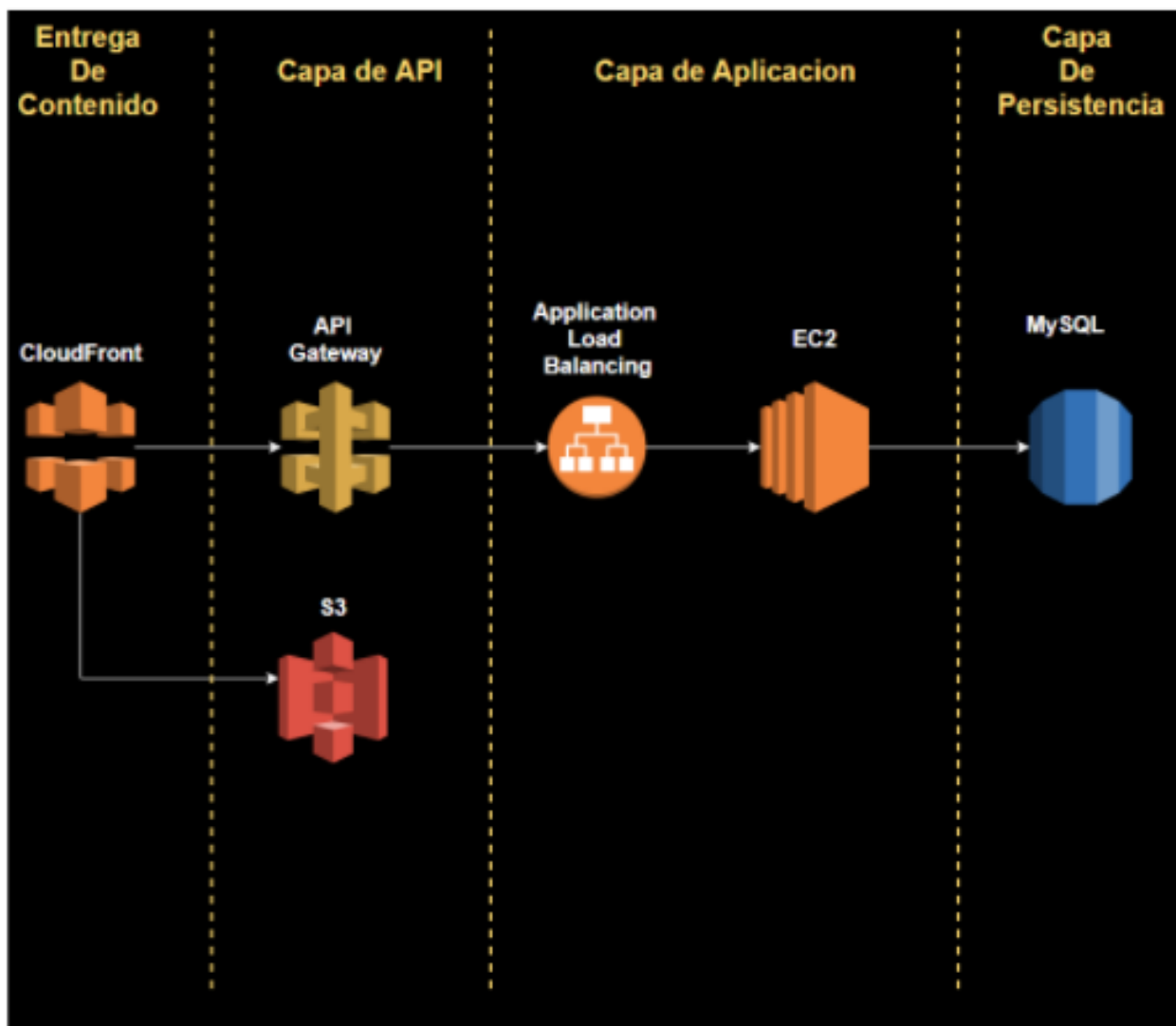


Figura 3: Diagrama de despliegue de microservicios en Amazon Web Services.

Esto es una típica implementación de una arquitectura de microservicios utilizando Amazon Web Services pero el costo mensual para poder funcionar es de alrededor 1100 dólares. Este es un costo aproximado ya que para tener una idea más acertada se necesitaría conocer información acerca del uso del sistema como podría ser el volumen de datos que van a ser guardados en la base de datos o la cantidad de tráfico web a la que el sistema será sometido.

## **ANÁLISIS DE VIABILIDAD COMERCIAL**

No corresponde dado que se enmarca en un proyecto UNDEX.

## **ANÁLISIS FINANCIERO**

No corresponde dado que se trata de un trabajo final de carrera.

## **ESTUDIO AMBIENTAL**

La elección de un servidor local o en la nube trae impacto ambiental negativo.

Mantener toda la información en la nube da la posibilidad de no tener que preocuparse por perder y/o procesar información ya que los proveedores de servicios en la nube permiten hacerlo desde cualquier lugar, pero esto significa tener la información en datacenters compuestos de miles de servidores que utilizan una cantidad de energía inmensa. Para mantener estos edificios se necesitan kilómetros de fibra óptica e infraestructura que requieren energía en el camino. En el centro, los datos se almacenan varias veces en discos duros, y la actividad constante de todos esos discos genera mucho calor, lo que requiere aires acondicionados que consumen mucha energía para proteger el equipo del sobrecalentamiento.

Además, los proveedores de servicios en la nube generan toneladas de desechos electrónicos, aunque algunos de estos proveedores, como Microsoft, están planeando proyectos de reciclaje y redistribución de los servidores y componentes degradados o en desuso.

Mantener un servidor local, permite controlar de manera más eficiente el uso de la energía y de los componentes, pero no se tiene la ventaja que provee la nube en el procesamiento y almacenamiento de los datos.

## **ESTUDIO SOCIAL**

La industria tecnológica y de desarrollo de software ha tenido un gran crecimiento en los últimos años. La Argentina no es ajena a este proceso: Actualmente la industria del software posee alrededor de 115.000 empleos y se proyecta para el año 2030 unos 500.000 en total o incluso más [?](Matias K, 2021).

En una industria en pleno crecimiento, la inserción de la mujer sigue presentando desafíos:

- Aunque en los últimos 15 años se duplicó la participación de mujeres en la industria de IT, todavía la brecha frente a la participación de hombres en esta misma industria es muy grande (70 % hombres, 30 % mujeres). [?](Mujeres en la industria del software, 2020)

- Solo el 30% de los estudiantes de carreras de Ciencia, Tecnología, Ingeniería y Matemática (CTIM) que se registran en universidades públicas y privadas son mujeres. [?](Ana inés & Cecilia, 2019)

En adición, la industria de Software es una de las pocas que ha mantenido su crecimiento durante los últimos años y posee uno de los mejores salarios en Argentina.

La brecha de género afecta también a la innovación y al crecimiento tanto del campo tecnológico como el desarrollo económico y social de un país. Los programas de inclusión vienen a tratar de disminuir esa brecha de género en tecnología, acercando experiencias formativas a mujeres adolescentes.

Es de suma importancia poder tener la posibilidad de realizar un seguimiento de cada alumno que hace estos programas de inclusión para poder sacar conclusiones, métricas y tomar decisiones con el fin de mejorar estas experiencias y por qué no, aumentar la cantidad de mujeres en el rubro de la tecnología.

## **EVALUACIÓN ECONÓMICA**

No corresponde dado que se trata de un trabajo final de carrera.

## Referencias

- [1] Resolución 441 de 2019 [Universidad de la Defensa Nacional]. Por la cual se establece el orden de los proyectos declarados como pertinentes y aprobados en la convocatoria 2019 del Programa UNDEX de la secretaría de Extensión. 23 de diciembre de 2019.
- [2] Matias, K. (2021). Economía del conocimiento. 13 de agosto de 2021, de Página12 website: <https://www.pagina12.com.ar/325647-software-el-pais-buscara-cuadruplicar-los-puestos-de-trabajo>
- [3] Mujeres en la industria del software. (2020). 11 de agosto de 2021, de cessi Argentina website: <https://www.cessi.org.ar/ver-noticias-en-los-ultimos-15-anios-se-duplico-la-participacion-de-las-mujeres-dentro-de-la-industria-del-software-2617>
- [4] Ana inés, V., & Cecilia, L. (2019). Un potencial con barreras: la participación de las mujeres en el área de ciencia y tecnología en Argentina. 13 de agosto de 2021, de Banco interamericano de Desarrollo website: [https://publications.iadb.org/publications/spanish/document/Un\\_potencial\\_con\\_barreras\\_la\\_participación\\_de\\_las\\_mujeres\\_en\\_el\\_área\\_de\\_Ciencia\\_y\\_Tecnología\\_en\\_Argentina\\_es\\_es.pdf](https://publications.iadb.org/publications/spanish/document/Un_potencial_con_barreras_la_participación_de_las_mujeres_en_el_área_de_Ciencia_y_Tecnología_en_Argentina_es_es.pdf)
- [5] Chicas en Tecnología. (2021) website: <https://chicasentecnologia.org/>
- [6] Kenneth S. Rubin. (2012). Essential Scrum: A Practical Guide to the Most Popular Agile Process (1ra ed.). Addison-Wesley
- [7] Erl, T. (2008). SOA Design Patterns. PRENCTICE HALL.
- [8] Nickoloff, J & Kuenzli S. (2014). Docker in action (2da ed). MANNING.