UNIVERSIDAD DE ANTIOQUIA

4NERS POINTS - HERRAMIENTA PARA FIDELIZACIÓN DE EMPLEADOS HACIA LA COMPAÑÍA (MODULO ESTADÍSTICAS)

Autor(es)

SERGIO ANDRÉS CASTRILLÓN VÁSQUEZ

Universidad de Antioquia

Facultad de Ingeniería, Departamento de ingeniería de sistemas

Medellín, Colombia

2019

4ners points - herramienta para fidelización de empleados hacia la compañía (modulo estadísticas)

Sergio Andrés Castrillón Vásquez

Informe de práctica como requisito para optar al título de:
ingeniero de sistemas.

Raúl Ramos Pollán.
Maria Fernanda Cubillos

Sergio Andrés Castrillón Vásquez - ingeniero de sistemas

Universidad de Antioquia
Facultad de Ingeniería, Departamento de ingeniería de sistemas.
Medellín, Colombia
2019.

Resumen

Actualmente el mercado de traspaso de personal en las empresas a nivel mundial se vive con mucha intensidad, no importa el área en la cual las empresas tengan su dominio (Software, psicologia, deporte, economía, etc), todas las empresas compiten y quieren tener en sus filas a las personas mas eficaces y más eficientes, las cuales sean capaces de generar el mayor valor posible a sus negocios y de hacerlos crecer en la industria en la cual están involucrados. Es por esto que la compañía S4N se dio cuenta de la necesidad tan grande que existe de darle a los empleados un ambiente agradable y gran cantidad de estímulos para recompensar su trabajo y sobre todo para lograr fidelizarlos a la empresa, todo con el fin de que, las personas, a la hora de tomar una decisión sobre su futuro como profesionales puedan poner en la balanza la gran compañía que tienen en frente, y esto les pueda ayudar a tomar la decisión de continuar y seguir desarrollándose profesionalmente en S4N.

Por lo mencionado anteriormente S4N le apostó a muchos proyectos de fidelización de empleados, uno de los cuales será el tema a tratar en este informe y sobre el cual se realizó un arduo trabajo en el periodo de prácticas académicas en el cuál estuve vinculado y pude aportar a su desarrollo. El proyecto se denomina 4ners Points - Calculadora de puntos, el cual consiste en una plataforma digital que permita generar bonificaciones a los empleados a medida que aumenta su tiempo con la compañía, es decir, mientras mas dures vinculado a S4N más beneficios obtendrás.

¿Cómo Funciona?

Como se mencionaba anteriormente, se busca que las personas perduren en la empresa y todo lo que aprendan profesionalmente lo puedan aportar al crecimiento de la misma. por ende se optó por crear unos bonos de diferentes valores y redimibles en diferentes lugares y almacenes a nivel mundial, dándole la oportunidad al trabajador de elegir en dónde quiere canjear sus bonos o también llamados puntos, existen muchas opciones para hacerlo, lo cual es garantía del éxito que tiene el proyecto y de la gran acogida por parte de los empleados, ya que se puede elegir entre diversas categorías como tecnología, educación, ocio,

transporte, mascotas y muchas más, y dentro de estas se puede elegir una gran variedad de almacenes para redimir esos puntos (Amazon, Netflix, centros comerciales en la ciudad, etc).

Cabe añadir que el proyecto se pensó para ser realizado una vez a al año, es decir, en el mes de diciembre se da a elegir los diferentes bonos a los empleados, los cuales se pueden repartir en diferente almacenes ya que existen dos categorías para dichos bonos, de trescientos mil y de quinientos mil pesos colombianos, y cada empleado tiene todo un año para hacer efectivo cada uno de los bonos que propiamente seleccionó.

También es importante señalar que el proyecto está pensado, como se mencionó anteriormente, para beneficiar a las personas que más perduran en la compañía, por tal motivo el valor de los bonos (cantidad de puntos) años tras año aumenta significativamente a cada uno de los empleados de S4N.

¿En qué se trabajo durante la práctica académica?

Este proceso de los 4ners points aún no se había visto reflejado en una plataforma digital que permita hacer automáticamente el cálculo de los puntos que le corresponden a cada empleado según la cantidad de tiempo que llevan vinculados a la compañía, por tal motivo, durante mi práctica académica me fue necesario entrar al equipo de trabajo que estaba abordando este proyecto y aportar a la adecuada realización de este.

Cuando ingrese ya se tenía gran parte de la funcionalidad core de la plataforma terminada sin embargo se me asignó un módulo dentro de la plataforma, el cual debía ser el encargado de mostrar la información, en forma de gráficas, que pasa por la calculadora de puntos (plataforma) y que le permita a la compañía saber cuales son las principales necesidades de sus empleados y donde estas sus principales intereses, y a partir de ahí tomar decisiones que permitan seguir conservando a los empleados partiendo de sus respectivos estilos de vida.

Introducción

Este proyecto consiste en el desarrollo de una plataforma interna que facilite y mejore significativamente la forma en la cual los empleados de la compañía puedan interactuar con los beneficios que S4N brinda al finalizar el año. Para ser más específico, la compañía año tras año entrega puntos (dinero en forma de bonos) a los empleados, los cuales pueden ser redimidos en tiendas como almacenes Éxito, Amazon, Netflix y muchas otras, todo esto con el fin de retribuir y premiar la fidelidad que tienen las personas que trabajan en S4N hacia la compañía, por tal motivo existen numerosas condiciones y muchas maneras de manejar los “4ners points”, donde finalmente se vio la necesidad de albergar toda esa funcionalidad en una plataforma de servicio que mejore la manera en la cual se estaba gestionando todo este proceso.

Al momento de iniciar mi práctica académica la plataforma se encontraba en un estado Beta, en donde su funcionalidad core (calcular los puntos correspondientes para cada empleado y la forma en que estos son canjeados) ya estaba operativa, sin embargo existía una gran deuda técnica en la plataforma y en otras funcionalidades que a la fecha no se tenían, como lo son administración de usuarios, generación de reportes, gráficas y estadísticas que brinden información asertiva y concreta de cómo se está manejando dicha campaña en la compañía.

Finalmente, algunos meses más tarde se logró cumplir el objetivo principal dentro de la plataforma que era adicionar el módulo de estadísticas, el cual fue explicado anteriormente, y además de ellos, se logró disminuir en gran medida la deuda técnica que tenía la plataforma, principalmente en su método de integración y despliegue continuo, el cual es un resultado muy satisfactorio ya que no se tenía contemplado como objetivo de la práctica académica pero se logró abordar y llevar a un feliz término, es la sección de anexos se hablará con más claridad del tema.

Cabe mencionar que el proceso de creación del nuevo módulo de estadísticas estuvo lleno de grandes retos tanto técnicos como humanos, principalmente al inicio de la práctica académica, ya que desconocía completamente la industria, de hecho, esta fue mi primera experiencia laboral, y por tal motivo me costó un poco acoplarme a las metodologías de trabajo que se usan en la compañía y las cuales explicaré a detalle a medida que avance con la lectura del informe.

Objetivos

Objetivo general:

Desarrollar un módulo dentro de la plataforma que permita generar estadísticas partiendo de la información y el flujo de datos a la que es sometida la misma, con el fin de poder generar valor y dar lugar a decisiones que permitan mejorar la permanencia de los empleados mediante campañas de fidelización hacia la compañía, dichas estadísticas deben dictar el camino a seguir por la compañía para la toma de decisiones, es decir, en estas se verán la cantidad de empleados que entran y salen, los proveedores favoritos por los empleados para canjear sus puntos, el tiempo de permanencia de los empleados en la compañía, etc.

Objetivos Específicos:

- Desarrollar herramientas visuales (gráficas) que permitan mayor legibilidad a la hora de leer las estadísticas.

- Reducir la deuda técnica que se tiene en la plataforma, haciendo más legible el código y mucho más escalable.

- Brindar información en tiempo real de los movimientos de la empresa en cuanto 4ners points (proveedores, convenios, empleados, etc).

Revisión de literatura (Marco teórico)

La revisión de la literatura se abordó desde dos frentes, uno muy de la parte de negocio (frente 1), es decir, el por qué es necesario un proyecto que apalanque el relacionamiento adecuado entre empleado y empresa, y otro frente que abarca la parte más técnica del proyecto (frente 2), como las tecnologías y las herramientas usadas y el por qué de dichas herramientas.

Frente 1 - Fidelización de empleados

Los consumidores son una parte fundamental en cualquier negocio, pero no es el único activo que hay que valorar. Muchas empresas cometen el error de centrarse únicamente en los clientes, dejando a un lado a los empleados y olvidándose de que sin ellos, los primeros no existirían.

Los trabajadores deben considerarse clientes internos, tan importantes como los externos y a los que hay que cuidar de la misma manera. Y es que son el fiel reflejo de la imagen de tu empresa y, la forma en que se sientan dentro de tu marca, será lo que transmitan de cara al exterior[1].

El proyecto 4ners points comprende diferentes módulos los cuales permiten sistematizar mediante la implementación de una plataforma desarrollada con tecnología de vanguardia el proceso en el cual se calcula la cantidad de puntos que debe ser asignada a cada empleado y la forma en la cual el empleado repartirá dichos bonos.

Es de aclarar que las estrategias de fidelización que están implementando las compañías actualmente son de admirar y cada vez se esfuerzan más por mantener un ambiente agradable entre el empleado y la empresa, por tal motivo se optó por esta plataforma, la cual apalanca con creces lo mencionado anteriormente, además de esto Cooper, B., Floody, B. & Mc Neill, G. (2003), expresan que *“Un cliente asiduo es publicidad gratuita... Mientras mejor conozca a sus clientes, más fácil le será darles el nivel de servicios y la calidad que ellos esperan*” (Cooper, Floody, Mc Neill, 2003, p. 178).

Frente 2 - Plataforma implementada para llevar a buen término el proyecto.

El stack tecnológico elegido para el desarrollo del nuevo módulo de estadísticas fue:

Scala:

Lenguaje de backend, el cual es combina programación orientada a objetos y funcional en un lenguaje conciso de alto nivel, los tipos estáticos de Scala ayudan a evitar errores en aplicaciones complejas, y sus tiempos de ejecución de JVM y JavaScript le permiten construir sistemas de alto rendimiento con fácil acceso a enormes ecosistemas de bibliotecas [2].

Angular:

Es un framework de desarrollo para JavaScript creado por Google. La finalidad de Angular es facilitarnos el desarrollo de aplicaciones web SPA y además darnos herramientas para trabajar con los elementos de una web de una manera más sencilla y óptima [3].

PostgreSQL:

Es un potente sistema de base de datos relacional de objetos de código abierto con más de 30 años de desarrollo activo que le ha ganado una sólida reputación de confiabilidad, solidez de características y rendimiento[4].

NPM:

Es el registro de software más grande del mundo. Los desarrolladores de código abierto de todos los continentes usan npm para compartir y pedir prestados paquetes, y muchas organizaciones usan npm para administrar el desarrollo privado también [5].

Slick:

Slick es una consulta moderna de base de datos y biblioteca de acceso para Scala. Le permite trabajar con datos almacenados casi como si

estuviera usando colecciones de Scala y al mismo tiempo le da control total sobre cuándo ocurre un acceso a la base de datos y qué datos se transfieren. Puede escribir las consultas de su base de datos en Scala en lugar de SQL, aprovechando así la comprobación estática, la seguridad en tiempo de compilación y la composición de Scala. Slick presenta un compilador de consultas extensible que puede generar código para diferentes backends[7].

Cabe aclarar que la idea inicial de utilizar estas herramienta no recayó directamente sobre mi, ya que como mencionaba anteriormente cuando ingresé al equipo ya se tenía gran parte de la plataforma funcionando con este stack tecnológico, por ende para realizar la funcionalidad core de mi pasantía (modulo de estadisticas) lo principal que tuve que hacer fue estudiar a fondo cada tecnología y ver cómo podría ayudarme a resolver la problemática dada, además de esto si se implementaron algunas librerías desde cero, las cuales fueron proveídas por NPM, dentro de las cuales destacó **angular-google-charts** ya que es la que permite visualizar toda la información de las gráficas de una manera organizada, coherente y que realmente genera valor al proyecto.

Metodología de desarrollo

Se implementó a lo largo del periodo de prácticas académicas una unión entre el marco de trabajo **Kanban**, el cual se define: “Un enfoque innovador para la gestión de proyectos, que permite que los equipos sean más efectivos y hacen que su proceso sea más ágil.” y el marco de trabajo **Scrum**, en el cuál se trata de planificar tus proyectos en pequeños bloques o Sprints, e ir revisando y mejorando el anterior.

Kanban permite alcanzar el objetivo del proyecto mediante la división de pequeñas tareas que sean visualizadas en un tablero y que permitan estar en tres columnas: “Para hacer”, “Haciendo” y “Hecho”, esto ayuda a progresar entre las actividades, contemplar el avance o las tareas que faltan para alcanzar la finalización del proyecto, generando el mayor valor posible al cliente en tiempos más cortos.

Mientras que por el contrario Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos, adicional a esto en Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales[6].

En el equipo al cual ingresé ya se había definido previamente la metodología con la cual se iba a trabajar en el proyecto, por tal motivo no pude ser partícipe de las decisiones en su momento, y razón por la cual estuve un poco perdido al iniciar mi periodo de prácticas, ya que era desconocido para mi el enfoque que se le daba al avance y las entregas de las historias de usuarios que estaban predefinidas en el roadmap de la plataforma.

El Sprint duraba dos semanas en la cuales se implementanban todas las ceremonias de la metodología Scrum (Review, planning, retro, refinamiento, y en algunas ocasiones se realizaban dailys), pero en lo personal, en las primeras semanas considero que no puede entregar casi valor al equipo debido a que me encontraba en etapa de “entrenamiento” y fue más o menos a partir de la tercera semana donde empecé a realizar historias de usuario, definidas junto con el PO y el equipo que empezarían a acercarme al cumplimiento de los objetivos planteados para mi práctica académica.

Interpretación y discusión de los resultados

Mi práctica académica la puedo dividir en 4 puntos muy importantes, los cuales detallo a continuación:

Aprendizaje:

Las primeras semanas me fueron asignadas historias de usuario que no contemplaban en lo absoluto el módulo de estadísticas, pero que fueron de vital importancia para entender la arquitectura actual de la plataforma, pude entender los fos microservicios que se manejaba, tanto el de backend como el de frontend, y además de esto empezar a aprender el stack de tecnología que se utilizaba, en cual introducía un nuevo paradigma de programación del lado del backend.

Adicional al código de backend que tuve que empezar a conocer y a entender se le sumó el framework de angular, el cual no fue muy complicado para mi porque en ocasiones pasadas ya había tenido la oportunidad de abordarlo, sin embargo si me fue muy útil este tiempo de aprendizaje para lograr entender las prácticas y los patrones que se seguían en la plataforma, y los cuales debía tener presentes a la hora de realizar el módulo de estadísticas.

Infraestructura:

Fue quizá la más importante mejora técnica que se logró realizar en la plataforma, y la cual tomó bastante tiempo implementar ya que la curva que aprendizaje que se requería era bastante, va de la mano con el proceso de despliegue y de integración continua de la plataforma, la cual hasta ese momento era nula.

Aunque como este no fue el objetivo general de la práctica académica, pero sin embargo se logró dejar funcionando adecuadamente y representó gran inversión de tiempo a lo largo de la pasantía se explicará a detalle en la sección de anexos, al final de este informe.

Backend:

Para el módulo de estadísticas se tuvo que tocar el microservicio de backend de la plataforma, para desde allí, poder exponer una numerosa cantidad de servicios con la información que se requiera mostrar en el front.

La arquitectura que se utilizó en este módulo fue un arquitectura por hexagonal, donde la principal característica es tratar de dejar el lenguaje de dominio puro,

utilizando algunos adaptadores a nuestro lenguaje como lo son los DTO y los DAO.

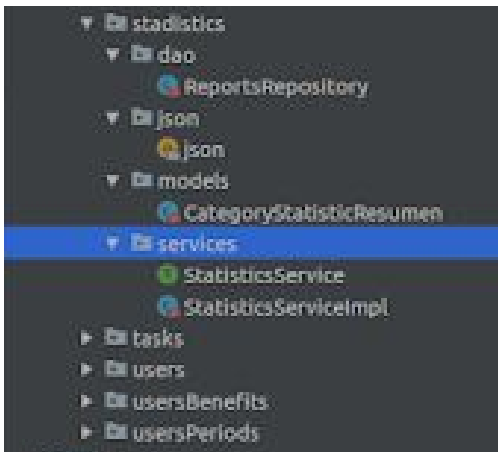

Fig 1. Estructura del módulo

Los DTO (data transfer object) y los DAO (data access object) son lo objetos que se utilizaron para mapear toda la información que se requiera de la base de datos y para dicho motivo se empleó Slick, el cual es una biblioteca de acceso a datos en scala que da la sensación de manejar elementos de bases de datos casi tan intuitivamente como si fueran colecciones de Scala, sin embargo, se debe ser muy cuidadoso para no dañar el dominio de nuestra aplicación con tipados que no corresponden a este, por tal motivo se implementaron estos tipos de patrones de arquitectura, cabe resaltar que en el paquete que he denominado DAO es donde se realiza ese proceso de “adaptación” del lenguaje de base de datos al lenguaje de dominio de la aplicación, todo esto para mantener limpio el paquete de servicios, el cuál es el encargado de orquestar toda la información y de dar una respuesta en un tipo propio del dominio, para más adelante ser el controlador quien se encargue de dar formato a la respuesta del servicio; para el caso del módulo de estadísticas todos los servicios de backend fueron expuestos en formato JSON.

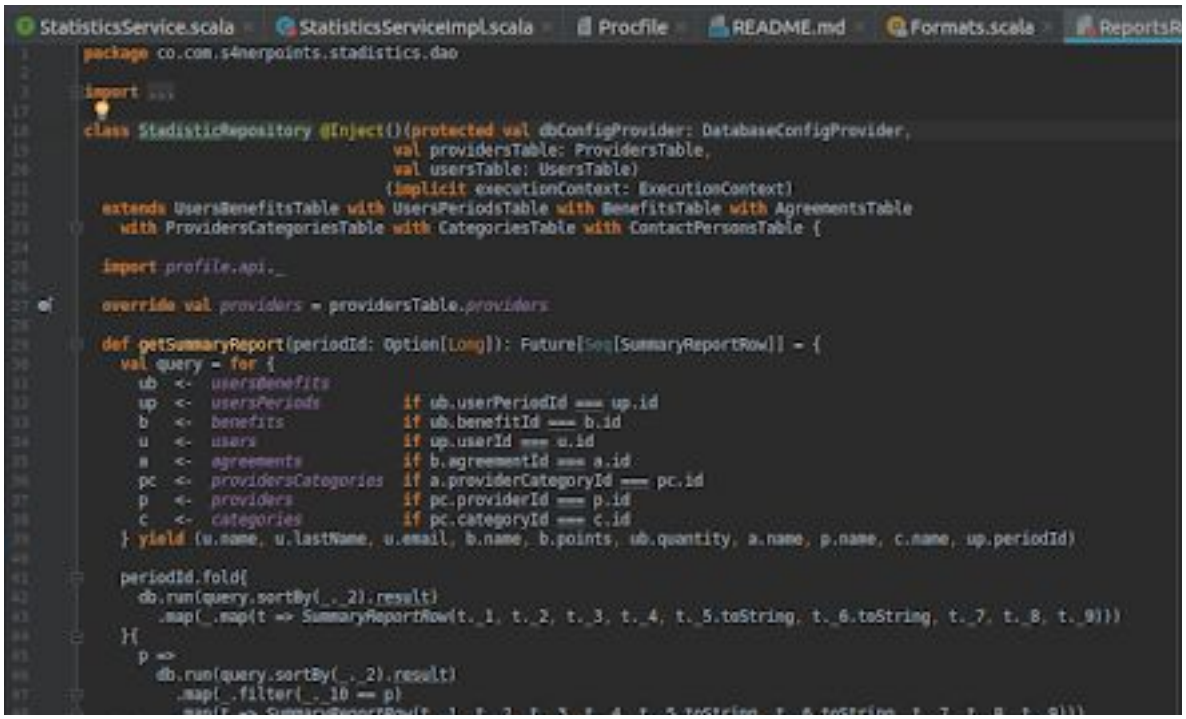

Fig 2. Ejemplo de código DAO

También para la implementación de los servicios del lado del backen se utilizó el patrón de service interpreter, el cuál consta de definir en archivos independientes el álgebra y la implementación, el álgebra es únicamente la firma de los métodos que corresponden a ese servicio y el interpreter es donde esas firmas tomarán valor, esta clase debe extender del álgebra sin embargo es completamente independiente a los tipados que se usen dentro de esta.

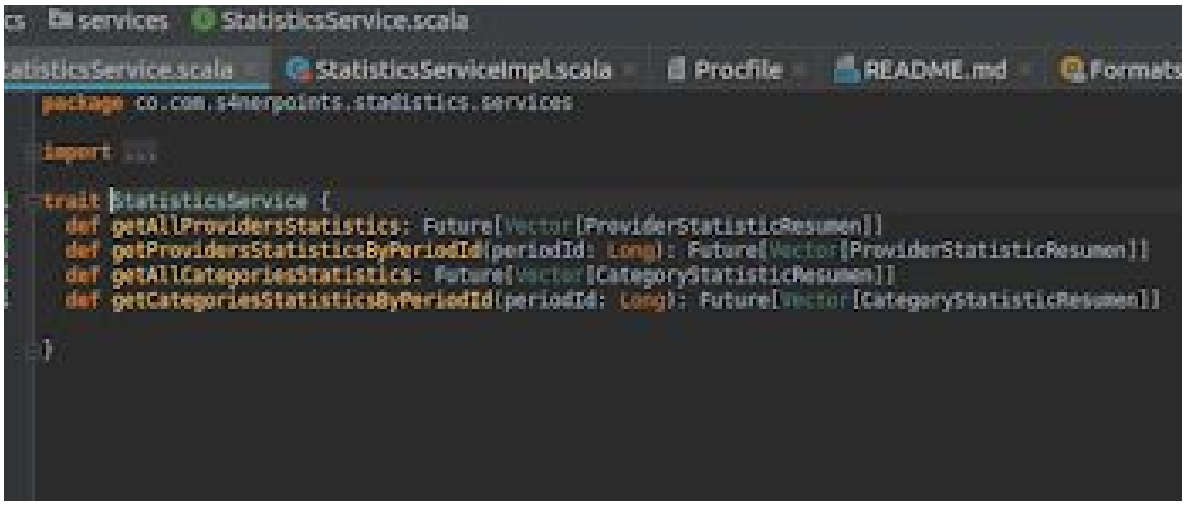

Fig 3. Patrón service interpreter

Una capa más arriba esta el controlador, el cual e es el encargado de orquestar el llamado a los servicios, los cuales están escritos en el lenguaje de dominio en cuestión, y es por tal motivo que es el encargado de parsear la respuesta a formato JSON (para el caso de modulo de estadisticas).

Fig 4. Controlador del módulo de estadísticas.

Finalmente se encuentra el archivo de rutas con el cuál serán expuestos nuestro servicios, es donde decidimos el path que tendrá cada uno de los servicios y el controlador que acciona la respectiva lógica que está detrás del llamado a este, la cual se resume en consultar información en base de datos, traerla a nuestro lenguaje de dominio por medio de DAO y DTO crear la lógica necesaria según el servicio para posteriormente parsear a JSON y exponer el servicio vía HTTP.

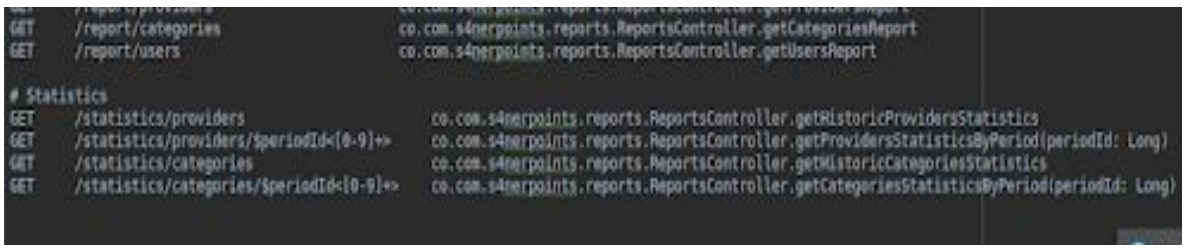

Fig 5. Rutas de los servicios del módulo de estadísticas.

Frontend:

La arquitectura del front es un poco más simple de entender porque en todo el proyecto se ha trabajado por los módulos que el mismo framework de angular proporciona, es decir, para la implementación del nuevo módulo de estadísticas, se puede decir que fue como crear un proyecto desde cero, ya que cree mi propio módulo y desde allí empecé a trabajar, sin embargo tuve en cuenta muchas de las prácticas de programación adoptadas por la plataforma en todos los diferentes módulos para trata de mantener la coherencia a la hora de leer el código en el microservicio.

Lo primero que se realiza del lado del front es la adaptación dentro del menú de la plataforma de nuevo módulo de estadísticas como se observa en la figura número 6.

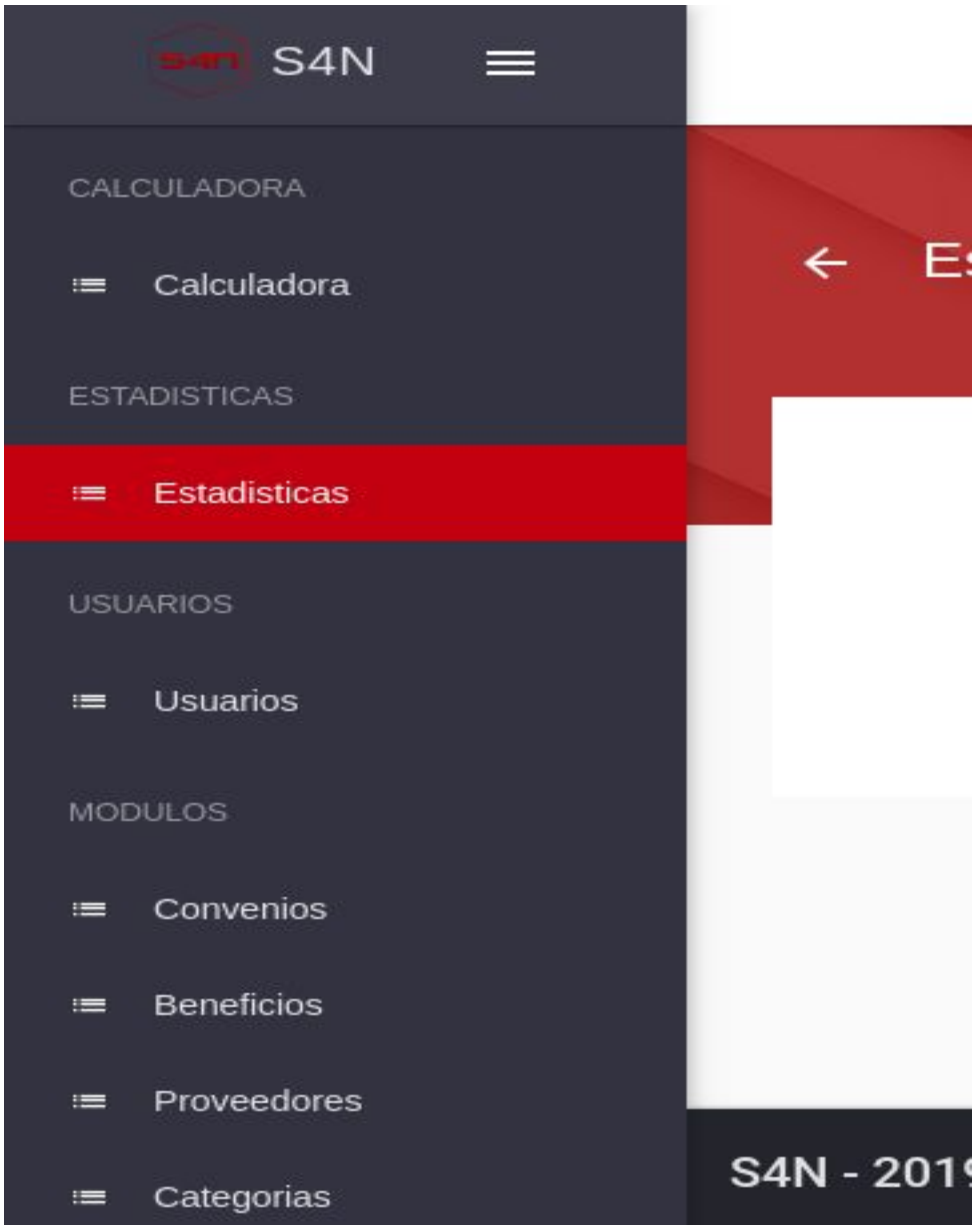

Fig 6. Menú de la plataforma

Después de tener expuestos en internet todos los servicios necesarios para que nuestro frontend pudiera pintar la información en la gráficas fue sencillo consumir esa información con la librería HttpClient de Angular.

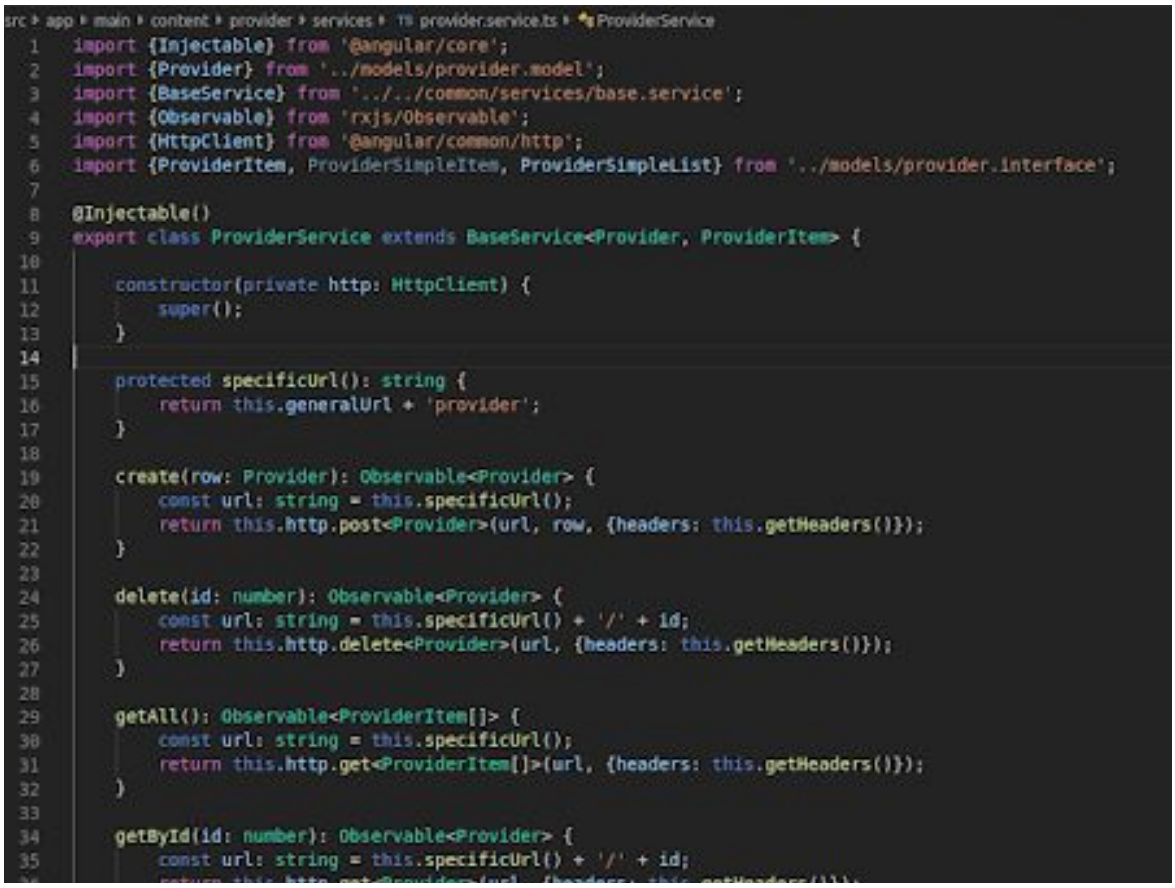

Fig 7. Ejemplo de consumo de servicios externos con HttpClient de Angular

Todo lo anterior para con la ayuda de la librería **angular-google-charts** se pudiera pintar la información consumida de una forma adecuada, bonita y organizada.

Cabe destacar que se realizaron diferentes servicios que dan información pertinente de cuales son los almacenes que prefirieron los empleados para canjear sus puntos, además de cuáles son las categorías en las cuales se invierten más bonos, y todo lo anterior se usa para el historial de la calculadora (desde que se empiece a utilizar) o para el periodo seleccionado, siendo un periodo lo mismo que un año, es decir, se pensó este módulo para saber a cuales años fueron mejores para X proveedor o simplemente saber cuál es el mejor proveedor a lo largo de la historia.

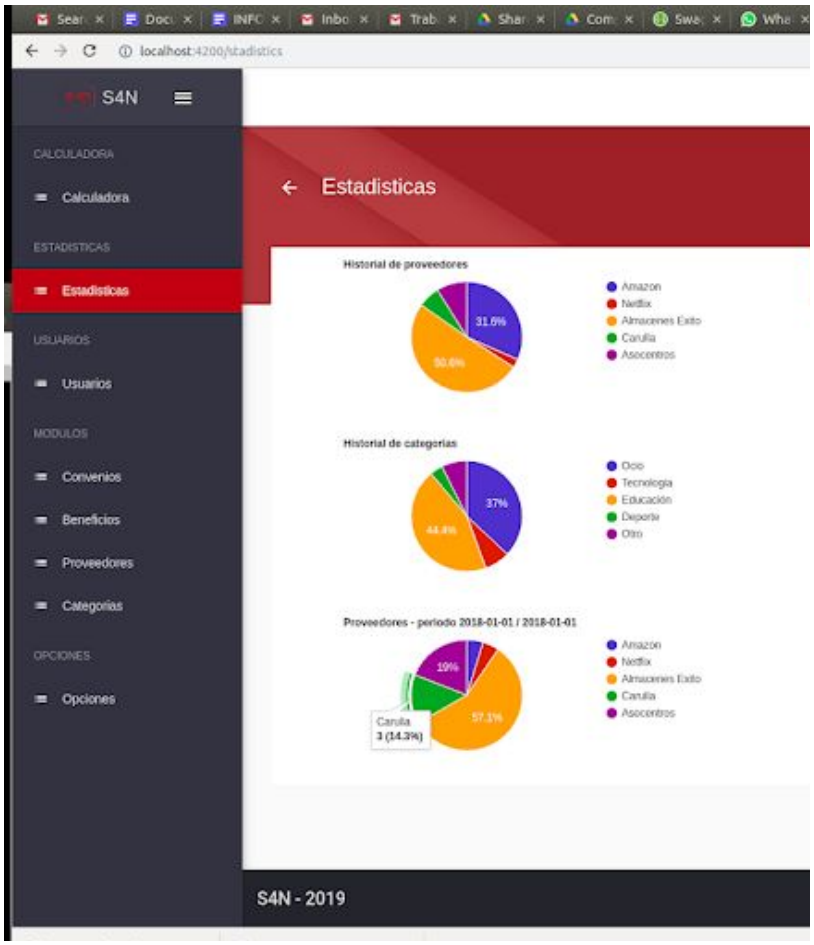

Fig 8. Ejemplo de estadísticas mostradas en pantalla.

Resultados y análisis

Los resultados del trabajo realizado en las prácticas académicas fueron satisfactorios puesto que se logró culminar a cabalidad el objetivo general y todos los objetivos específicos, sin embargo la plataforma tiene “mucha tela por cortar”, puesto que se alcanzó a realizar un breve estudio sobre la factibilidad de empezar a realizar módulos que contengan nuevas funcionalidades que permitan la fidelización del cliente bajo el mismo modelo de calculadora de puntos, pero esta vez no únicamente mirar la cantidad de tiempo que las personas cumplen en S4N sino también la calidad de su trabajo, las habilidades blandas y muchas otras

competencias que hacen a las personas grandes profesionales y que es objetivo al fin y al cabo para la empresa es explotar cada uno de esas habilidades en cado uno de sus trabajadores.

Además, técnicamente se tuvieron algunas reuniones para estudiar la factibilidad de implementar machine learning sobre la plataforma, con el fin de ayudar a las personas de talento humano de la compañía a tomar decisiones sobre cuál es el rumbo más adecuado para cada uno de los trabajadores dentro de la compañía.

Conclusiones

Respecto a los objetivos planteados inicialmente y al ver el desarrollo del trabajo, se puede decir que se cumplieron los objetivos de la práctica académica satisfactoriamente.

Adicionalmente, otras conclusiones que nos permite expresar el proyecto son:

- El trabajo con herramientas corporativas es algo delicado porque algunas componentes son privados y la idea no es que salgan de la compañía.

- Se logró mostrar en pantalla de una forma organizada, intuitiva y amigable la información que se requiere para empezar a tomar decisiones que permita generar un mejor ambiente de trabajo para los empleados.

- La práctica académica es el primer acercamiento al campo laboral, donde se pueden ver conceptos y definiciones adquiridas en el campo académico.

- Las metodologías ágiles permiten un dinamismo en la solución de software respecto a la corrección de errores y a la refactorización del pensado inicial y la generación de valor para con el negocio

- Se logra brindar información exacta del estado actual de la calculadora cada que un empleado registra sus bonos.

- Se logró desplegar en producción la plataforma con una arquitectura serverless, lo cual era una gran deuda técnica del proyecto, ya que anteriormente estaba desplegada en un servidor físico de la compañía.

Bibliografía

[1] (2017, diciembre 12). ¿Por qué apostar por la fidelización de empleados? - Blog de .... Se recuperó el octubre 14, 2019 de https://blog.gestazion.com/por-qu%C3%A9-apostar-por-la-fidelizaci%C3%B3n-de-empleados

[2] (n.d.). Download - Scala. Se recuperó el febrero 19, 2019 de https://www.scala-lang.org/download/

[3] (2017, agosto 5). ¿Que es Angular y para qué sirve? - Victor Robles | Victor Robles. Se recuperó el febrero 19, 2019 de https://victorroblesweb.es/2017/08/05/que-es-angular-y-para-que-sirve/

[4] (n.d.). PostgreSQL. Se recuperó el febrero 19, 2019 de https://www.postgresql.org/

[5] (n.d.). About npm | npm Documentation. Se recuperó el febrero 19, 2019 de https://docs.npmjs.com/about-npm

[6](n.d.). Qué es SCRUM - Proyectos Ágiles. Se recuperó el octubre 14, 2019 de https://proyectosagiles.org/que-es-scrum/

[7] (n.d.). Slick - Lightbend. Se recuperó el octubre 14, 2019 de http://slick.lightbend.com/

Anexos

Infraestructura

Para la infraestructura se crearon tres imágenes de docker (Frontend, backend y base de datos) las cuales se pusieron a correr en tres contenedores diferentes, esto funciona en cualquier ambiente que se corra, es decir, en la máquina local de la persona que desee ejecutar el proyecto o el producción, que son instancias de los microservicios desplegadas en un cluster de EC2 en AWS , creadas y ejecutadas con la ayuda de templates de Terraform, que evita que la configuración sea manual y engorrosa en AWS.

El proceso de Integración y despliegue continuo se implementó mediante el mismo repositorio de código donde está albergado el proyecto, Gitlab fue la herramienta encargada de este proceso ya que nos permite configurar los pipelines de una forma facil y rapida, ademas que únicamente requiere de algunos pasos como la creación de las tres imágenes, loguearse con la cuenta de AWS donde está toda la infraestructura de la compañía y hacer push sobre un repositorio ECR.

Las imagen de docker para el microservicio de back se logró fácilmente con el comando **sbt docker:publishLocal** el cuál busca el archivo sbt.build de la aplicación y busca todas la dependencias que este requiere para su correcto funcionamiento y las monta todas en la imagen junto con el proyecto, para que únicamente quede correr la imagen en un contenedor.


/bin/bash
/bin/bash
2019-10-04 08:05:21 sergio in ~/Documents/Estudio/s4nerpoints-back
→ sudo sbt docker:publishLocal
[sudo] password for s4n:
[info] Loading settings from plugins.sbt ...
[info] Loading project definition from /home/s4n/Documents/Estudio/s4nerpoints-back/project
[info] Updating {file:/home/s4n/Documents/Estudio/s4nerpoints-back/project/}s4nerpoints-back-build...
[info] Done updating.
[warn] Found version conflict(s) in library dependencies; some are suspected to be binary incompatible:
[warn] * org.webjars:webjars-locator-core:0.33 is selected over 0.32
[warn] +- com.typesafe:npm_2.12:1.2.1 (depends on 0.32)
[warn] +- com.typesafe.sbt:sbt-web:1.4.3 (scalaVersion=2.12, sbtVersion=1.0) (depends on 0.32)
[warn] * org.codehaus.plexus:plexus-utils:3.0.17 is selected over {2.1, 1.5.6}
[warn] +- org.apache.maven:maven-settings:3.2.2 (depends on 3.0.17)
[warn] +- org.apache.maven:maven-repository-metadata:3.2.2 (depends on 3.0.17)
[warn] +- org.apache.maven:maven-aether-provider:3.2.2 (depends on 3.0.17)
[warn] +- org.apache.maven:maven-model:3.2.2 (depends on 3.0.17)
[warn] +- org.apache.maven:maven-core:3.2.2 (depends on 3.0.17)
[warn] +- org.apache.maven:maven-artifact:3.2.2 (depends on 3.0.17)
[warn] +- org.apache.maven:maven-settings-builder:3.2.2 (depends on 3.0.17)
[warn] +- org.apache.maven:maven-model-builder:3.2.2 (depends on 3.0.17)
[warn] +- org.sonatype.plexus:plexus-sec-dispatcher:1.3 (depends on 1.6.1)
[warn] +- org.eclipse.sisu:org.eclipse.sisu.plexus:0.0.0.M5 (depends on 2.1)
[warn] * com.google.guava:guava:23.0 is selected over {10.0.1, 16.0, 18.0}
[warn] +- io.methvin:directory-watcher:0.3.2 (depends on 23.0)
[warn] +- com.fasterxml.jackson.datatype:jackson-datatype-guava:2.8.8 (depends on 18.0.1)
[warn] +- org.eclipse.sisu:org.eclipse.sisu.plexus:0.0.0.M5 (depends on 18.0.1)
[warn] +- com.spotify:docker-client:8.9.0 (depends on 18.0.1)
[warn] Run 'evicted' to see detailed eviction warnings
[info] Loading settings from build.sbt ...
[info] Set current project to s4nerPoints (in build file:/home/s4n/Documents/Estudio/s4nerpoints-back/)
[info] Packaging /home/s4n/Documents/Estudio/s4nerpoints-back/target/scala-2.12/s4nerpoints_2.12-1.0-sources.jar ...
[info] Done packaging.
[info] Wrote /home/s4n/Documents/Estudio/s4nerpoints-back/target/scala-2.12/s4nerpoints_2.12-1.0.pom
[info] Main Scala API documentation to /home/s4n/Documents/Estudio/s4nerpoints-back/target/scala-2.12/api...
[info] Packaging /home/s4n/Documents/Estudio/s4nerpoints-back/target/scala-2.12/s4nerpoints_2.12-1.0-web-assets.jar ...
[info] Done packaging.
[info] Compiling 112 Scala sources and 12 Java sources to /home/s4n/Documents/Estudio/s4nerpoints-back/target/scala-2.12/classes ...
model contains 368 documentable templates
[info] Main Scala API documentation successful.
[info] Packaging /home/s4n/Documents/Estudio/s4nerpoints-back/target/scala-2.12/s4nerpoints_2.12-1.0-javadoc.jar ...
[info] Done packaging.
[info] Done compiling.
[info] Packaging /home/s4n/Documents/Estudio/s4nerpoints-back/target/scala-2.12/s4nerpoints_2.12-1.0.jar ...
[info] Done packaging.
[info] Packaging /home/s4n/Documents/Estudio/s4nerpoints-back/target/scala-2.12/s4nerpoints_2.12-1.0-sans-externalized.jar ...
[info] Done packaging.
[info] Sending build context to Docker daemon 107.1MB
[info] Step 1/6 : FROM openjdk:8
[info] ---> 0bbedcf8366c
[info] Step 2/6 : WORKDIR /opt/docker
[info] ---> Using cache
[info] ---> 4ebf8b65ed67
[info] Step 3/6 : ADD --chown=daemon:daemon opt /opt
[info] ---> c8fe3448a8f3
[info] Step 4/6 : USER daemon
[info] ---> Running in 3a7f56b8bb3d
[info] Removing intermediate container 3a7f56b8bb3d
[info] ---> 5bdf7b67bd49
[info] Step 5/6 : ENTRYPOINT ["bin/s4nerpoints"]
[info] ---> Running in 9e0ff5861150
[info] Removing intermediate container 9e0ff5861150
[info] ---> 5170f185e6e9
[info] Step 6/6 : CMD []
[info] ---> Running in ae4fe9f71ce9
[info] Removing intermediate container ae4fe9f71ce9
[info] ---> cf3b8389534e
[info] Successfully built cf3b8389534e
[info] Successfully tagged s4nerpoints:1.0
[info] Built image s4nerpoints:1.0
[success] Total time: 57 s, completed Oct 14, 2019 8:06:48 AM
[INFO] [10/14/2019 08:06:48.648] [Thread-2] [CoordinatedShutdown(akka://sbt-web)] Starting coordinated shutdown from JVM shutdown hook
2019-10-04 08:06:48 sergio in ~/Documents/Estudio/s4nerpoints-back

Fig 9. Creación de imagen de docker de un proyecto de Scala.

La imagen del frontend se logró con el archivo tradicional de dockerfile, que es el encargado de tener la receta que requiere el componente para la creación y el correcto funcionamiento de la imagen de docker.

Fig 10. DockerFile FrontEnd

Finalmente tenemos la imagen de la base de datos, la cual está en postgreSQL, para ello se necesito crear un dump y por ende un contenedor de la base de datos, Dicha imagen parte de hacer pull de una imagen ya existen de postgreSQL 9.6 del docker hub y luego de la configuración de sus variables de ambiente (Usuario, password, nombre de la bd, etc)

Después de tener las tres imágenes creadas es posible unirlas usando un docker-compose, que es un archivo yaml que contiene la receta para ejecutar las tres imagenes a mismo tiempo, y que únicamente basta con correr el comando **docker-compose up -d**, para levantar las tres imágenes y tener la plataforma corriendo en la cualquier máquina con un solo comando.

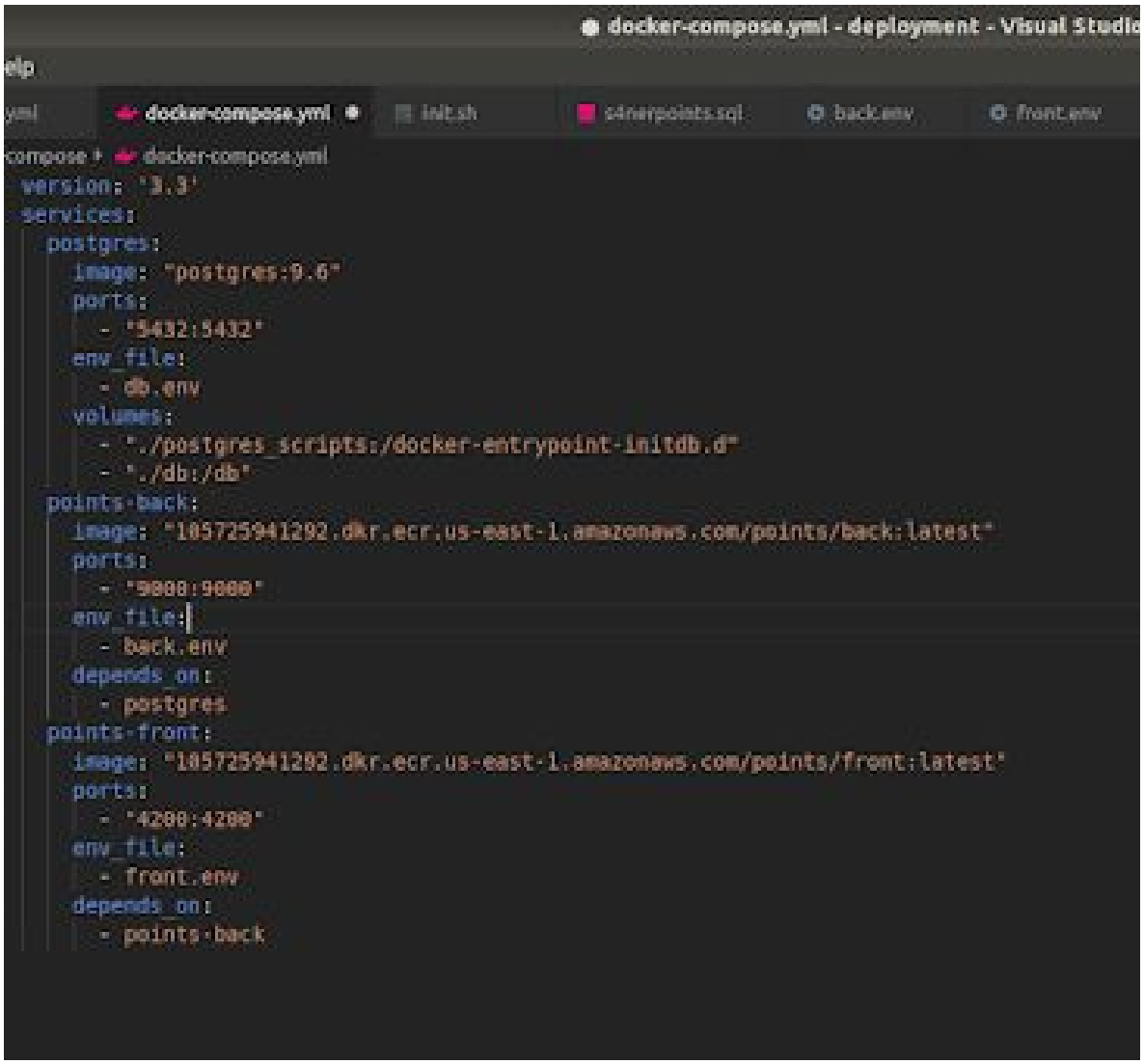

Fig 11. Docker-compose de la calculadora de puntos.