



Wiki – Proyecto Integrador Final

Equipo 6

“Proyecto HOMU”

Integrantes:

- Annichini, Florencia.
- Contreras, Leonel.
- Mora Lizcano, Joseph Ian.
- Pérez, Micaela.
- Ramírez, Victoria.
- Rivero Tirado, Juan Diego.

Contenido

Homu Project	3
Link al Slide de presentación en Demo	3
Equipo.....	4
Floencia Annichini	4
Juan Diego Rivero Tirado	5
Micaela Pérez.....	6
Victoria Ramírez.....	7
Leonel Contreras.....	8
Ian Joseph Mora Lizcano	9
Metodología de Trabajo	10
Explicación de la metodología de trabajo y puntos positivos y/o negativos.....	10
Asignación de roles y repartición de tareas.....	11
Otros acuerdos.....	11
Bitácora del Proyecto.....	11
Sprint 1.....	11
Sprint 2.....	12
Sprint 3.....	14
Sprint 4.....	15
Tecnologías Utilizadas	16
Gestión del proyecto	16
Desarrollo de proyecto.....	16
Documentación Técnica.....	17
Ambiente de desarrollo.....	17
Bases de datos	18
Componentes y diagramas de la aplicación.....	19
Infraestructura	19
Testing.....	21

HOMU PROJECT

Proyecto HOMU busca llevar adelante una plataforma de servicios web de reservas de distintos tipos de alojamiento en cualquier parte de Argentina y Colombia, con una potencial ampliación de los servicios ofrecidos y maximizando el alcance, siempre desde una perspectiva didáctica, comunicacional y tecnológica.

Nos pusimos como objetivo principal el aprendizaje durante el desarrollo del proyecto y el trabajo en equipo como herramienta fundamental para avanzar con las ideas propuestas. Buscamos potenciar nuestras habilidades tanto técnicas como blandas, colaborando entre todos para generar una dinámica de grupo acorde a las necesidades que iban surgiendo y los cambios que debían realizarse, siempre atentos a las nuevas ideas que podían surgir para mejorar y potenciar nuestro proyecto.

Nuestro proyecto se llevó a cabo de forma integral en cuanto a su desarrollo haciendo un seguimiento exhaustivo de cada parte del mismo y haciendo uso de la nube de Amazon Web Service (AWS) como plataforma de despliegue del proyecto, siempre teniendo en cuenta los recursos disponibles.

Link al Slide de presentación para la Demo:

<https://docs.google.com/presentation/d/1c9a8YZlkgrc9nWQrm1x0IPFYsH8PXXIPrjbJpWcYAu0/edit?usp=sharing>
















EQUIPO

• Nombre del integrante, una breve descripción o biografía del integrante comentando qué conocimientos tenía al iniciar la carrera, una breve descripción de los roles desempeñados y las tareas realizadas durante los diferentes sprints y una tabla indicando las fortalezas actuales en las principales áreas de tecnología de la carrera.

Maria Florencia Annichini

• Al iniciarme en este nuevo desafío no contaba con ninguna herramienta o habilidad en programación, pero sí con la convicción y las ganas de aprender de este nuevo mundo. De a poco a lo largo de los bimestres fui aprendiendo e incorporando distintas habilidades técnicas que me permitieron llegar a mis conocimientos actuales pero que a la vez me dan la motivación para querer seguir aprendiendo de forma continua y adecuándome a las nuevas tecnologías que puedan surgir y a lo que me demanden futuros proyectos o trabajos.

• A lo largo de los cuatro sprints me desarrollé en el rol de Base de Datos de forma principal, siendo esta rama de la programación una de las que más interés me despierta y que gracias al apoyo de mi equipo pude explotar. Durante el primer sprint también participé del desarrollo del backend con el modelado de las distintas capas del modelo vista-controlador que implementamos. En el segundo sprint tuve participación tanto en bases de datos como en backend continuando con la misma línea de lo hecho en el primer sprint, en el tercero continúe con mi papel en base de datos, en menor medida hice aportes al backend, más que nada siempre siguiendo la misma línea y tuve una participación del lado del testing implementando Jest. En el cuarto sprint seguí la misma línea que en el tercero ocupando mi rol en base de datos definiendo la configuración final, el modelo, el DER y la conexión a AWS, en backend principalmente hice un aporte a el manejo de excepciones y aportando al testing con Jest.

Tecnología	Fortaleza
Frontend	  
Backend	  
Infraestructura	  
Testing / QA	  
Bases de Datos	  






Juan Diego Rivero Tirado

- Cuando inicie la carrera no tenía ningún conocimiento previo a la programación ni nada relacionado con el desarrollo de software, desde el primer bimestre hasta el último, logre aprender muchas cosas incluso de forma autónoma ya que había temas que no lograba comprender del todo, pero todo esto me ayudó a poder prepararme, y lograr aprender y llegar al lugar que estoy ahora y me asombra todo lo que he logrado aprender en un año ya que esto es hasta ahora el inicio de una formación profesional en donde iré adquiriendo nuevas experiencias que me ayudaran en próximos proyecto.
- Durante el primer sprint tuve el rol de Frontend y de Backend, y a partir del segundo sprint solo estuve en el rol de Frontend, a pesar de que no tenía mucho conocimiento de React logré aprender bastante y lograr manejar distintos lenguajes.

En el primer Sprint en el back empezamos a desarrollar las entidades y el modelado, y en el Frontend empezar a crear la página de inicio, la de registro e inicio de sesión. En el segundo sprint estuve con el rol del Frontend en donde tuvimos que agregarle nuevas vistas como la del detalle del producto, nuevos apartados y nuevos estilos.

En el tercer sprint en el Frontend se empezó a realizar las conexiones con el Backend y a implementar funcionalidades e implementamos opcionales.

En el cuarto sprint ya teníamos que consumir todo la API para poder llamar a los endpoints requeridos para lograr enviar los datos en la base de datos y verlos reflejados en el Frontend e implementamos los opcionales.

Tecnología	Fortaleza
Frontend	
Backend	
Infraestructura	
Testing / QA	
Bases de Datos	






Micaela Pérez

- Al iniciar la carrera tenía conocimientos básicos de lógica de programación, sabía usar if y poco más. Mi recorrido durante los bimestres fue bastante orgánico, solía entender bien el contenido de Playground y las clases, y generalmente intentaba buscar mini proyectos para aplicarlos y practicar de forma autónoma. La carrera de CTD me ayudó a conseguir cosas que veía muy lejanas. Aprendí muchísimo en poco tiempo gracias a como está planteada la experiencia y, realmente, me asombra ver todo el avance y que, ese conocimiento nuevo, es realmente funcional y aplicable.
- Durante los cuatro sprints me desarrollé en el rol de Frontend. Quise tomarlo como un desafío ya que el Backend era en donde me sentía más segura y más conocimientos tenía al respecto.

En el primer Sprint, desde el Front, nos encargamos de desarrollar los templates, la mayoría aún sin funcionalidad a excepción de las rutas. Luego, en el segundo, comenzamos con los eventos de estos templates ya realizados tales como algunos filtros de búsqueda desde la página principal.

En el tercer Sprint tuvimos que conectar la información que mostraba nuestra página, con los endpoints de las APIs que habían hecho nuestros compañeros/as desde el Backend, así como también poder guardar nuevos datos, que eran generados por el usuario.

Llegando al final, en el cuarto Sprint, tuvimos que realizar las conexiones para que un usuario Admin pueda crear un producto nuevo. También implementamos bastantes opcionales como eran los de la información relacionada al Covid-19 del usuario, los mails al registrarse y reservar, el listado de mis reservas y el listado de mis favoritos.

Tecnología	Fortaleza
Frontend	
Backend	
Infraestructura	
Testing / QA	
Bases de Datos	

Victoria Ramírez






• Al comenzar la carrera, no tenía ningún conocimiento previo de programación, sólo conocía un poco de HTML y CSS por algunos cursos online que había tomado por pura curiosidad. Durante el recorrido del año, gracias al primer track de la carrera, pude avanzar más allá de lo que hubiera imaginado, dándome cuenta que puedo aplicar mis conocimientos en los entornos que sean necesarios.

• Durante los sprints en general, el principal rol que desarrollé fue Testing y como secundario en Backend. En el primer sprint, se empezó a realizar el armado de planillas y los entornos de ambiente en Postman para poder comenzar a probar las APIs.

En el segundo sprint, se siguió desarrollando la planilla, sumando dos columnas más de automatización y suites dado los requisitos de las issues. A su vez, también comenzó a testearse el Frontend con Jest y las nuevas APIs desarrolladas en Postman con variables y ambientes nuevos. También desarrollé ciertas modificaciones en el Backend dado que haciendo el testeo se encontraron defectos.

En el tercer sprint, empecé a tomar mayor participación en Backend al realizar la parte de Spring Security y correcciones por errores encontrados. Siguió testeándose en Postman para verificar también los tokens creados con su verificación y autenticación. En Testing se siguió completando las planillas y aportando en Jest brevemente.

En el cuarto y último sprint, se siguió en Backend realizando modificaciones y correcciones. En la parte de Testing, se culminaron las planillas y las colecciones en Postman. También se realizaron pruebas con Selenium dado que ciertas librerías de React causaban inconvenientes al testearlo en Jest, el resto se realizó de forma unitaria en la carpeta de Jest.

Tecnología	Fortaleza
Frontend	
Backend	
Infraestructura	
Testing / QA	
Bases de Datos	

Leonel Contreras

• Al iniciar la carrera no terminaba de comprender el mundo al cual me adentraba, no tenía conocimientos o habilidades previas en programación. Lo único con lo que contaba era con el entusiasmo de empezar y las ganas de aprender en el transcurso de la cursada.

Durante la cursada pude adherir los conocimientos básicos sobre distintas áreas e ir puliendo las habilidades técnicas mientras avanzaban los bimestres. Gracias a la modalidad de la carrera, pude adquirir una gran cantidad de conocimiento en muy poco tiempo con ayuda de los profesores y un gran acompañamiento de toda la camada. Estoy seguro que seguirá de la misma manera en el segundo tramo.






• Durante el primer Sprint me desarrollé en el lado del Frontend, pero a partir del segundo fui rotando y ayudando entre las tareas del Backend, Testing, Base de Datos y un poco en Infraestructura.

En el primer sprint, nos encargamos de los template de la página de inicio casi sin funcionalidades, pero enfocándonos en los estilos.

En el segundo sprint tuve una pequeña participación en Infraestructura, y en mayor parte en backend con el desarrollo de entidades y modelado, y automatización de test en la parte de Testing.

En el tercer sprint me mantuve en la misma área, continuando haciendo aportes al Backend, Base de Datos y automatizando tests con Jest del lado del Testing.

En el último Sprint, terminamos de hacer modificaciones en el Backend y de automatizar tests con Jest y Selenium IDE en Testing.

Tecnología	Fortaleza
Frontend	
Backend	
Infraestructura	
Testing / QA	
Bases de Datos	

Ian Joseph Mora Lizcano






- Inicie la carrera con muchas expectativas de aprendizaje ya había pasado por cursos similares y no lograba convencerme la metodología de aprendizaje, sin embargo, al llegar a DH todo fue diferente y la forma de aprender fue bastante dinámica, los profesores hicieron que los conceptos complejos fueran aterrizados de una manera más sencilla.

- A lo largo de los sprints estuve apoyando en los roles de Backend e Infraestructura, de los cuales el que mayor reto representó fue Infraestructura, familiarizarme con todas las tecnologías de AWS, Terraform y Gitlab debido a la falta de conocimiento. Sin embargo, con investigación y algunas citas con el TL logramos resolver y llegar al final con todo lo que se nos pedía por parte del Product Owner.

Dentro de las tareas desarrolladas en Infraestructura estuvieron el diseño, creación y administración de los recursos mediante la consola de AWS y también a través de la infraestructura como código con Terraform, por último, la automatización de todo el proyecto haciendo uso GitLab, aplicando CI/CD, mediante la creación de pipelines dentro de cada rama de trabajo en Git.

En tanto a lo que se refiere al Backend pude aplicar los conocimientos adquiridos en los bimestres pasados, con gran fortuna pude desarrollar parte del proyecto apoyando en temas como la creación de algunas entidades y sus respectivas clases e interfaces que conforman la arquitectura MVC.

Adicionalmente, puse en práctica algunos conceptos de SQL aplicados con Hibernate para poder implementar la lógica de algunos de los filtros que solicitaban para el proyecto a través de HQL y Spring Boot. Finalizando con la implementación de una librería para el uso de avatar para el perfil de usuarios registrados en la web.

Tecnología	Fortaleza
Frontend	
Backend	
Infraestructura	
Testing / QA	
Bases de Datos	

METODOLOGÍA DE TRABAJO

- Breve explicación de la metodología de trabajo utilizada (Scrum), indicando los puntos que les hayan resultado positivos y / o negativos.

Utilizamos a lo largo de los distintos sprints la metodología Scrum como base organizacional del trabajo a realizar y la designación de los roles de cada integrante del equipo a lo largo del proyecto.

Se desarrolló de manera tal que cada sprint comenzaba con la lectura grupal del mismo, su posterior validación y luego encuentros organizados como Daily donde el grupo compartía de forma individual el avance logrado, lo que estaba en desarrollo y las dificultades que se pudieran tener y la forma en que se buscaría resolverlo o hacerlo de forma colaborativa con otro integrante del grupo.

También se fueron haciendo Weeklys, una vez a la semana, donde nos juntábamos con el Product Owner para comentar el avance del proyecto y las posibles dudas que nos habían surgido a fin de entender las necesidades del cliente y sus objetivos para el proyecto.

Se hicieron Review donde se presentaba el MVP alcanzado al final de cada sprint y donde se fue teniendo una devolución constructiva, y luego una Retro donde se ponía de manifiesto lo que creíamos que nos impulsaba a seguir avanzando y qué cosas podríamos mejorar en vistas de lo que nos iba quedando de trabajo por realizar.

Esta metodología organizacional nos sirvió como equipo para ir manteniendo un conocimiento continuo de cómo estaba cada integrante respecto al desarrollo del proyecto y sus tareas asignadas. De esta forma siempre que surgieron problemas se pudieron ir resolviendo más allá de lo individual y de forma colaborativa para avanzar de forma más eficiente, aunque si bien nos ayudó mucho desde lo grupal para estar todos conectados con cada parte del proyecto, por momentos esto hizo que queramos abarcar más allá de nuestras tareas y los roles asignados sean menos estrictos en cuanto a quien llevaba a cabo su ejecución.

- **Explicación de cómo manejaron la asignación de roles dentro del equipo (si los hubo) y cómo se organizaron a la hora de repartir las tareas.**

La asignación de roles se llevó a cabo de forma eficiente y teniendo en cuenta los objetivos a nivel de aprendizaje de cada integrante y sus fortalezas.

Designamos de forma interna un Scrum Master que fue la persona que se encargó de ir ordenando las tareas y manteniendo una planificación ordenada del desarrollo del proyecto y la implicancia en el mismo de cada integrante.

En general se logró una buena organización de los mismos, aunque en la mayoría de los casos nos manejamos en forma de bloques de trabajo donde dentro de estos asignamos las tareas a desarrollar tanto de forma individual como conjunta con otro integrante del equipo.

- **Opcionalmente pueden describir otros acuerdos a los que hayan llegado para trabajar en equipo, gestionar el control de versiones, normalizar el código producido, etc.**

Buscamos organizarnos en GitLab creando Branchs Per Environment, asignadas cada una a un integrante del equipo para su actualización. Además, cada integrante tenía su rama donde iba subiendo sus avances de forma regular para que todo el equipo tuviera acceso.

También organizamos reuniones en plataformas de Meet o Zoom para mantenernos actualizados y otras herramientas como Discord para dejar toda la información necesaria que ayudara a nuestro proyecto.

BITÁCORA DEL PROYECTO

Sprint 1:

- **Frontend (Micaela, Leonel y Juan Diego):**

- Se implementó React Router.
- Se eligieron paleta de colores, fuentes, logo, entre otros. Se buscó acomodar queries para mobile first.
- Se hizo corrección de bugs.
- Se implementaron templates de login, la correcta ubicación del logo, botones y menú hamburguesa, y un sitio responsive.

- **Backend (Florencia, Joseph y Juan Diego):**

- Organizamos horarios, subtareas, entre otros para comenzar.
- Creamos la API de la entidad de categoría.
- Se concluyó con todo lo pedido en este sprint de backend y en general funcionaba y ya se podía conectar con el frontend.

- **Testing (Victoria):**

- Se hizo el armado de templates de testing y una planilla en notion con tareas hasta tener el Gitlab habilitado.
- Se concluyeron los casos de prueba de las historias de usuario.
- Se comenzó a investigar sobre pruebas exploratorias.
- Testeo del frontend y backend de lo realizado hasta el momento.

- **Infraestructura (Victoria):**

- Se comenzó a buscar herramientas para la realización del boceto teniendo en cuenta las consideraciones necesarias para un correcto diseño del mismo.
- Se obtuvo la información de usuarios y contraseñas para AWS, por lo que se comenzó a investigar sobre su correcto uso.
- Se finalizó con el boceto sobre el modelo de la infraestructura que íbamos a estar usando a lo largo del proyecto.

- **Base de datos (Florencia):**

- Creación del sprint de la BD para MySQL.
- Inicialmente teníamos una base de datos corriendo en h2 asociada al backend e hicimos el cambio a MySQL.

Sprint 2

- **Frontend (Micaela y Juan Diego):**

- Se terminó lo del Login pendiente del Sprint 1 y cosas de estética y funcionalidad que estaban pendientes.
- Se conectaron las Apis del Backend de Ciudades, Categorías y Productos.
- Se crearon Cards según considerasen para mayor organización.
- Se comenzó a estilar el Producto Detail.
- Se siguió haciendo la conexión con el backend.
- Se comenzó a estilar la página de productos.

• **Backend (Florencia, Joseph, Leonel y Victoria):**

- Se crearon todas las tablas nuevas.
- Se empezaron a realizar los mapeos.
- Se actualizaron algunas clases del proyecto para acceder a los filtros de categorías y ciudades desde producto.

• **Testing (Victoria, Leonel y Florencia):**

- Se empezó a actualizar las planillas para incluir lo del Sprint 2.
- Se actualizó en GitLab el JSON con los tests de Postman de API Categorías realizado con variables (Sprint 1).
- Se testeó el Login pasado del Front.
- Se continuó con los testeos a los endpoint.
- Se terminaron de hacer la planilla con los casos de pruebas, defectos, testeos exploratorios y ejecuciones.
- Se hicieron los test de Postman para verificar que todo funcione y también se vea desde el frontend.
- Seguimos con el uso de Jest para realizar pruebas.

• **Infraestructura (Joseph y Leonel):**

- Se hizo el bucket S3 para subir las imágenes.
- Se crearon las redes y subredes para almacenar las BBDD.
- Investigación sobre los Pipelines y el montado del Front en la Infra, y se comenzó a trabajar sobre esto.
- Se hizo un deploy de prueba de frontend.

• **Base de datos (Florencia):**

- Se crearon todas las tablas pertenecientes a la épica 2 en MySQL.
- Se armó también un diagrama de las mismas tablas y sus relaciones.
- Se hicieron las relaciones y el script de todo.

Sprint 3

• Frontend (Micaela y Juan Diego):

- Se estilo y se hizo responsive el calendario.
- La página y botón de reserva se puso en funcionamiento y se arrancó con el tema de los filtros en el inicio de sesión.
- Se arregló el responsive de la galería de imágenes para desktop y se la conecto a la API.
- Se terminaron todas las issues que no dependían del back.
- Se conectó el register y el login con el backend.
- Se está consumiendo el token.

• Backend (Victoria, Leonel, Florencia y Joseph):

- Se logró configurar el Web Token y obtener la respuesta correspondiente.
- Se hizo un authentication controller a modo de prueba.
- En el package model se hizo una carpeta de autenticación para el request y response.
- Se agregó una query en el repository de user para que puedan funcionar las pruebas.
- Se armó un controller de autenticación nuevo para que todo pueda ser testeado por Postman.
- Se realizaron los entity, dto, service, controller y repository correspondientes a las mismas.
- Se constató que la base de datos conecta bien con el backend.
- Se actualizaron los filtros de buscar productos por ciudades, buscar productos por categorías y buscar reservas por productos.

• Testing (Victoria, Leonel y Florencia):

- Se hizo una colección a modo de prueba para el token y se armaron dos request (POST y GET) para testearlo.
- Se hizo una nueva colección de prueba para ingresar usuarios y roles.
- También se armó otros requests para comprobar que el token está creado y permite el ingreso autenticado.
- Se verificó por medio de las pruebas que las request dieran el status asignado en las issues.
- Se estuvo a más del 60% del testeo de los componentes con Jest.

- **Infraestructura (Joseph):**

- Se investigó como conectar el Postman con la VPC.
- Se realizó el deploy de backend y del frontend
- Se agregó un dominio para la página en general (<http://homu.online/>).
- Se pudo arreglar lo que rompía la conexión con la BD, ahora se puede ingresar desde la IP pública.

- **Base de datos (Florencia):**

- Se crearon las tablas de roles, usuario y reserva, y se hicieron las conexiones correspondientes.
- Se hicieron las relaciones y el script de todo.
- Se actualizó y se cargó la BD con categorías, ciudades, características, productos e imágenes mediante Postman.

Sprint 4:

- **Frontend (Micaela y Juan Diego):**

- Se terminó de realizar la conexión a la API en login y reservas para que sea funcional a como expresaban las issues.
- Se conectó el template reserva exitosamente.
- Se terminó de realizar la conexión a la API en login y reservas para que sea funcional a como expresan las issues con la actualización de back (que login funcione con email).
- Se arreglaron bugs que estaban presentes.

- **Backend (Victoria, Joseph, Florencia, Leonel):**

- Se actualizó el spring security para que funcione con email.
- Se agregó el filtro de fechas y fechas y ciudades.
- Se agregaron tablas nuevas para el bloque de políticas de front.
- Se realizaron los cambios en mapeos necesarios para que funcionen en post de productos.
- Se agregaron las url en spring security para los métodos nuevos de las tablas agregadas.

• **Testing (Victoria, Florencia y Leonel):**

- Se crearon colecciones nuevas que engloban lo que se viene trabajando en back y front.
- Se crearon excepciones para agregar test negativos en postman en métodos delete by id y find by id
- Se terminaron de testear los nuevos métodos agregados y los filtros de fechas y fechas y ciudades.
- Se hizo el testeo de funcionalidades con Selenium IDE y testeos con Jest.
- Se crearon carpetas en la colección para verificar el correcto funcionamiento de las tablas nuevas.
- Se realizó el informe final de testing.

• **Infraestructura (Joseph):**

- Se realizaron actualizaciones a los cambios realizados de front y back.

• **Base de datos (Florencia y Victoria):**

- Se realizaron actualizaciones a las nuevas tablas creadas.
- Se realizaron actualizaciones en los diagramas y los scripts para llenar la BD.

TECNOLOGÍAS UTILIZADAS

Dentro de las tecnologías utilizadas durante el proyecto estas se podrían dividir según el área de desarrollo:

• **Gestión del proyecto:** para esto usamos tanto GitLab como repositorio principal y como principal gestor de issues por sprint, también acá fuimos anotando la información obtenida de cada Daily y Weekly.

Usamos GitHub como repositorio alternativo de uso interno según nuestro orden en bloques por entorno de desarrollo, y un canal de Discord donde se iban anunciando las novedades por área del proyecto y donde colaborativamente fuimos agregando herramientas o ideas para el desarrollo del mismo.

• **Desarrollo del proyecto:** dentro de los IDE's utilizados estuvieron Visual Studio Code para trabajar sobre el frontend principalmente y IntelliJ idea para el backend.

Como sistema de control de versiones se usó Git ya que es una de las mejores

herramientas disponible y es compatible con sistemas y protocolos que fuimos usando como HTTP, SSH y es una forma eficaz de manejar proyectos.

Para la virtualización del proyecto se usó Amazon Web Services.

Dentro de las tecnologías utilizadas dentro de cada área de desarrollo están:

- **Frontend:** HTML5, CSS3, React, uso de librerías y uso de conexiones a API's.
- **Backend:** Spring Boot, Java, JUnit, log4j, maven, Spring Boot Security y sistema de conexión vía Hibernate para el mapeo de la base de datos y la conexión a la misma.
- **Base de datos:** MySQL Workbench, diagramas DER, UML y Diagrams.io.
- **Infraestructura:** AWS, creación de gestor de recursos S3 para el frontend, una VPC conformada por una public subnet que contiene la instancia EC2 creada y una private subnet Amazon RDS donde se desplegó la base de datos. Uso de terraform y pipelines.
- **Testing:** Postman, planilla de defectos, planilla de casos de prueba, Jest y Selenium IDE.

DOCUMENTACIÓN TÉCNICA DEL PROYECTO

• **Ambiente de desarrollo:** para comenzar cada integrante se creó una cuenta en GitLab. Allí se crearon de forma grupal dentro de un repository main las distintas Branchs Per Enviroment dentro de las cuales incluimos Base de Datos, Testing, Backend, Frontend e Infraestructura.

Por otra parte, cada integrante creó su rama donde fue subiendo los avances diarios antes del chequeo final y estar apto para hacer el merge con las ramas principales correspondientes.

Se definieron ámbitos de trabajo de forma colaborativa, Visual Studio Code, IntelliJ Idea, MySQL y Postman siendo necesario que cada integrante del equipo tenga descargados los mismos ya que en estos se fue trabajando y avanzando con el proyecto tanto de

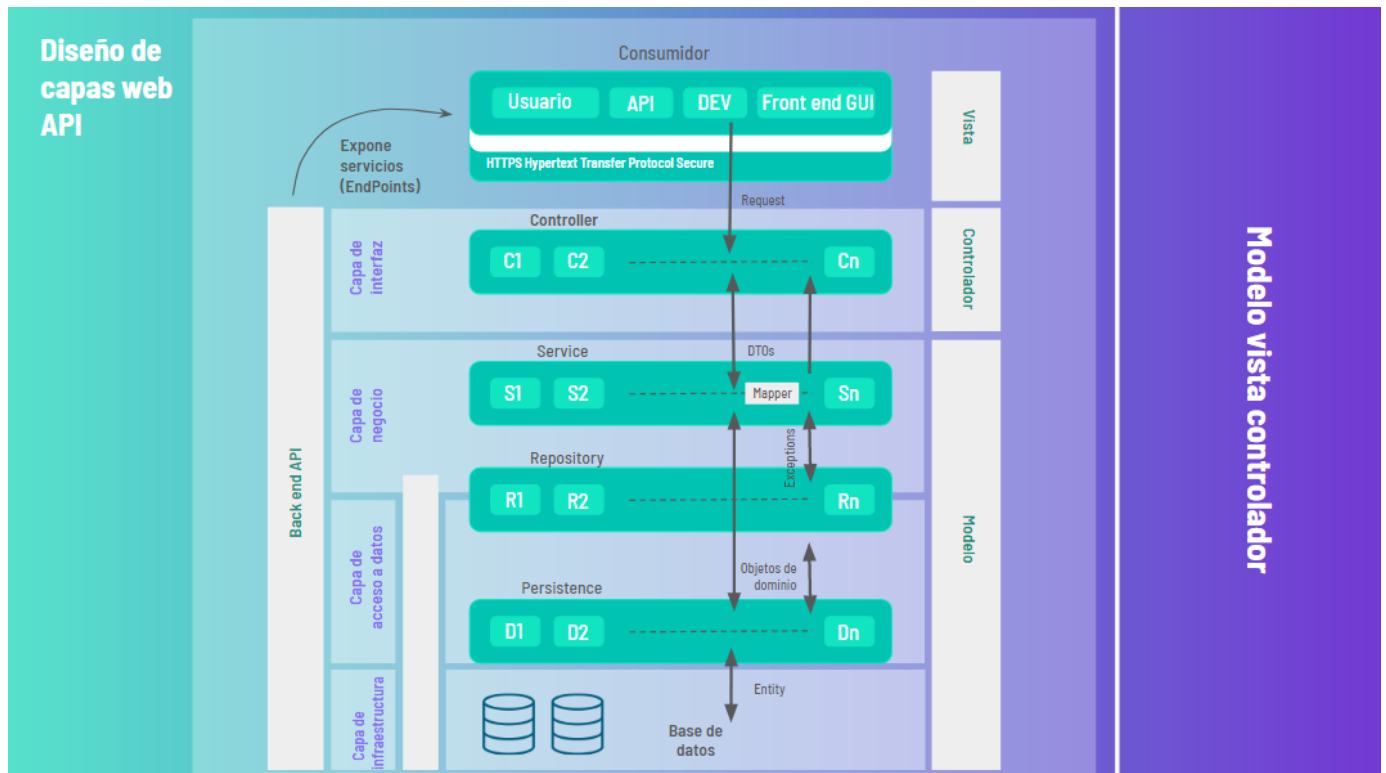
manera individual como conjunta previo a cada commit a realizar y a fines de ir completando las issues que se iban agregando.

• **Base de datos:** Se creó una base de datos de tipo relacional en MySQL Workbench para lo que se siguieron las distintas etapas:

- **Análisis de los requisitos del proyecto:** Conforme éste iba creciendo y manteniendo la posibilidad de que sea escalable y que sea capaz de solucionar problemas o llevar adelante propuestas que se puedan ir agregando. Se analiza las necesidades del cliente basándonos en lo aportado por el Product Owner. Se hizo una investigación de sistemas similares para poder tener una idea general de las necesidades que iban a poder ir surgiendo a medida que avanzábamos en el proyecto.
- **Identificación de las entidades a desarrollar:** En nuestro caso se crearon doce entidades finales (Booking, Cancellation, Category, City, Favourite, Feature, Image, Product, Role, Rule, Safety, y User).
- **Organización y creación de las tablas:** Tabulamos las entidades identificadas, trabajamos sobre las relaciones entre las mismas y su cardinalidad conforme a las exigencias del producto a desarrollar y en concordancia con lo que se necesitaba desde el frontend. Se identificaron y revisaron las claves foráneas generadas en las tablas.
- **Estandarización de las tablas:** Se buscó no tener atributos redundantes y optimizar los mismos según las necesidades que iban apareciendo o que se iban modificando, pero siempre sin sacrificar rendimiento de la misma. Con Reverse Engineer de la base de datos creada se creó el diagrama DER final.
- **Cargado de la Base de Datos:** se hizo la carga de los datos a las tablas desde Postman, la generación del script correspondiente y la exportación de la base de datos creada. Esta base de datos se fue probando desde el localhost antes de ser llevada a AWS para su implementación final en el servidor y para su uso en la página web que desarrollamos.

- **Componentes y diagramas de la aplicación:**

Diagrama de toda la aplicación con conexiones desde el Frontend, Backend, Infraestructura y Bases de Datos.



- **Infraestructura:** para la el despliegue de infraestructura partimos usando la herramienta de infraestructura como código terraform, la cual nos permitió organizar por módulos cada uno de los recursos que necesitábamos en el proyecto y realizar de manera más eficiente el despliegue asegurando el principio de la idempotencia (obtener siempre el mismo resultado ejecutando el mismo script) adicionalmente abre la posibilidad de automatizar el despliegue mediante el uso de pipelines dentro de gitlab.

Dichos recursos son los que podemos apreciar en el diagrama final, iniciando por la parte de redes, habilitamos una vpn con 3 subnet públicas, las cuales dos de ellas están en zonas de disponibilidad distintas como lo requiere el RDS (BD). Configuramos las tablas de ruteo para cada subred y habilitamos un punto de acceso a internet para la instancia EC2.

Se crean dos grupos de seguridad uno para BD y otro para la instancia que contendrá el Backend. También creamos una instancia t2 micro dentro de la capa gratuita y le

asignamos la respectiva subred y grupo de seguridad.

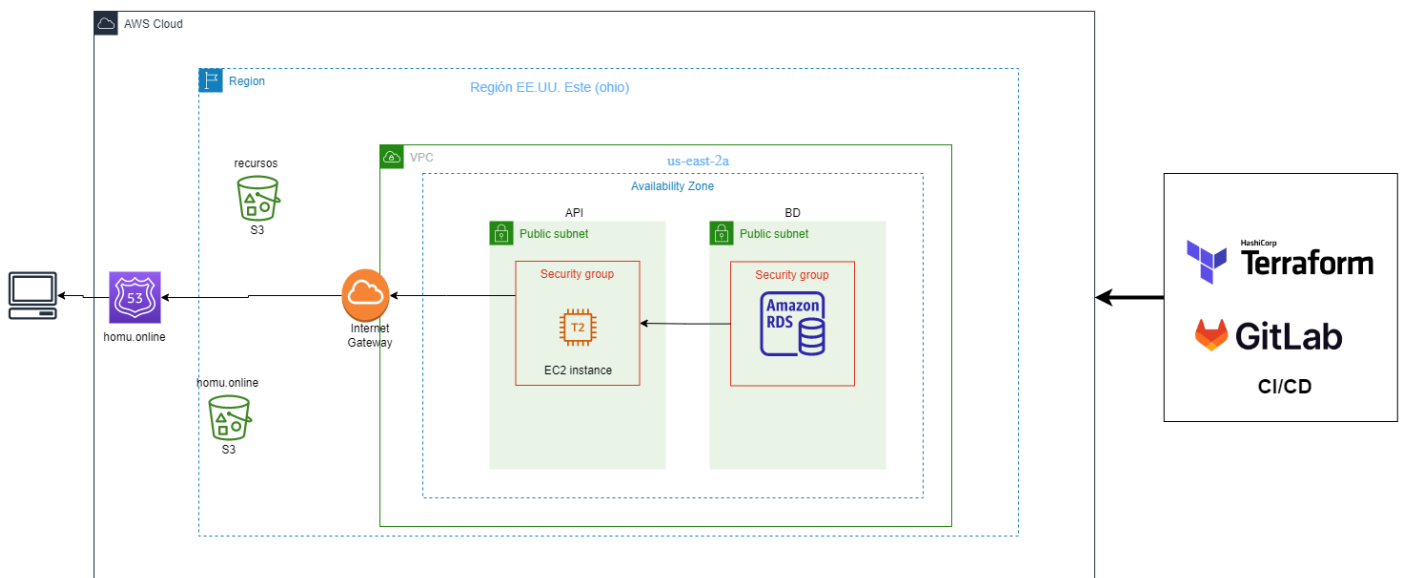
Le habilitamos las reglas de entrada por SSH y el puerto 8080 y salida a todo el mundo. Creamos dos buckets en S3, uno para los recursos de la aplicación como imágenes, logos, etc. y el otro para alojar el Front end. A cada uno de estos buckets se le configuran políticas con el fin de que sean públicos y accesibles.

Dentro de Route 53 configuramos el DNS del bucket que contiene el Frontend para obtener una url más amigable.

Antes de crear la BD, dentro de RDS debemos configurar un grupo de subredes seleccionando 2 de las subredes creadas anteriormente que tengan diferentes zonas de disponibilidad. Enseguida creamos la BD configurando el motor de bd en este caso MySQL y su versión, realizamos algunos otros ajustes adicionales como las credenciales de acceso, entre otras.

Creamos la BD y obtenemos el punto de enlace para llevarlo a las configuraciones del API, con sus respectivas credenciales y nombre de la base de datos.

Por último, realizamos CI/CD mediante gitlab a través de un archivo yaml configuramos cada uno de los jobs que nos permitan construir, testear y desplegar nuestra aplicación tanto para el back como para el front.



- **Testing y calidad:**

- **Link al reporte de Testing:** <https://drive.google.com/file/d/1i5KTsCx8-wnZ6WpHhtVlv1mMNp6sNPwm/view?usp=sharing>.
- **Entorno a Testing:**

Se organizaron ambientes para realizar las pruebas y poder testear las APIs de forma más fácil y también poder llenar automáticamente la BD a través de Postman.

Desde Jest se tomaban los componentes para testearlos con tests unitarios que ayudaban a la automatización con ciertos casos de pruebas. También se pudo automatizar con Selenium aquellas partes que por haber incongruencia con las librerías de React, dado los nuevos cambios, rompían en Jest y. por ende, optamos utilizar tal alternativa.

Por último, desde las planillas, se armaban los casos de pruebas según el backlog de del sprint para que contengan los requisitos y funcionalidades necesarios. En caso de encontrar defectos, se seguía iterando los casos de pruebas pertenecientes a dicha funcionalidad o issues hasta se solucionaran.