**Logotipo, nombre de la empresa

Descripción generada automáticamenteForma

Descripción generada automáticamenteDesarrollo del proyecto**

*Procesos de Ingeniería del Software*

Curso 2023-2024

**Autor:** Alejandro Navarro de la Cruz

Contenido

[Introducción y objetivos del Proyecto 3](#_Toc149231282)

[Sprint 1: Arquitectura base y puesta en marcha 3](#_Toc149231283)

[Reunión de preparación del Sprint 3](#_Toc149231284)

[Desarrollo del Sprint 3](#_Toc149231285)

[Reunión de revisión 4](#_Toc149231286)

[Reunión de retrospectiva 4](#_Toc149231287)

[Sprint 2: Usuarios 5](#_Toc149231288)

[Reunión de preparación del Sprint 5](#_Toc149231289)

[Desarrollo del Sprint 5](#_Toc149231290)

[Reunión de revisión 8](#_Toc149231291)

[Reunión de retrospectiva 8](#_Toc149231292)

[Anexos 9](#_Toc149231293)

[Diagrama de secuencia: Agregar Usuario 9](#_Toc149231294)

[Diagrama de secuencia: Inicio de Sesión en Local 9](#_Toc149231295)

Índice de figuras

[Ilustración 1: Modelo de dominio inicial 3](#_Toc149231304)

[Ilustración 2: Diagrama de arquitectura inicial 4](#_Toc149231305)

[Ilustración 3: Imagen PBI del Sprint 1 4](#_Toc149231306)

[Ilustración 4: Gráfica de Burndown del sprint 1 5](#_Toc149231307)

[Ilustración 5: Autenticación con GitHub 6](#_Toc149231308)

[Ilustración 6: Modal de Google One Tap 6](#_Toc149231309)

[Ilustración 7: Diseño de Inicio de sesión 7](#_Toc149231310)

[Ilustración 8: Diseño de registro de usuarios 7](#_Toc149231311)

[Ilustración 9: Diagrama de arquitectura sprint 2 8](#_Toc149231312)

[Ilustración 10: Gráfica de Burndown del sprint 2 8](#_Toc149231313)

[Ilustración 11: Diagrama de secuencia AgregarUsuario 9](#_Toc149231314)

[Ilustración 12: Diagrama de secuencia InicioSesionLocal 9](#_Toc149231315)

# Introducción y objetivos del Proyecto

En este documento se presenta el desarrollo de un proyecto para la asignatura de Procesos de Ingeniería del Software cuyo objetivo principal es desarrollar una aplicación basada en servicios con registro e inicio de sesión de usuarios.

En las siguientes secciones, se detallarán los aspectos más esenciales del proyecto separados por Sprint ya que el proyecto tiene un enfoque ágil.

# Sprint 1: Arquitectura base y puesta en marcha

## Reunión de preparación del Sprint

En este sprint, se comenzó a poner en marcha el proyecto, escogimos como herramienta de modelado StarUML y en ella creamos dos diagramas que comentaré más adelante. Y se empezó a diseñar e implementar la arquitectura base del proyecto.

Creamos un repositorio en el conocido software de gestión de proyectos, Jira y definimos dos épicas, desarrollo y documentación, de las cuales colgarán las historias asociadas a cada épica respectivamente.

Detallamos las tareas relacionadas con las historias y estimamos las horas para intentar llevar un control unificado.

## Desarrollo del Sprint

Durante el desarrollo del Sprint hemos ido cambiando la solución para ir puliendo la solución inicial. Primero implementamos la lógica, las pruebas y el modelo, luego el backend correspondiente a los artefactos que integraremos (nuestro modelo y las pruebas) en el servidor. Expusimos la lógica en la capa Rest e implementamos un cliente de comunicación con dicha capa.

Finalmente implementamos el componente de visualización para mostrar los formularios y desplegamos nuestra aplicación en Cloud Run (<https://arquitectura-base-github-5rfb3lj4yq-ew.a.run.app/>).

Modelamos el diagrama inicial de dominio que tendrá nuestra aplicación, el cual tendrá un Sistema que tendrá 1 o muchos Usuarios. Inicialmente Usuario solo tiene un Nick, pero posteriormente en siguientes Sprints lo modificaremos según nuestras necesidades.

Texto

Descripción generada automáticamente con confianza media

Ilustración : Modelo de dominio inicial

También modelamos el diagrama de la arquitectura de nuestra aplicación que contará con una arquitectura MVC por parte del cliente con esa capa de comunicación con el API Rest, para mantener desacoplados los datos y las funcionalidades de nuestro sistema. Y un servidor NodeJS clásico de tres capas, aunque la conexión con MongoDB se ha quedado fuera del alcance de este Sprint.

Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración : Diagrama de arquitectura inicial

## Reunión de revisión

Tras realizar la reunión de revisión con el PO concluimos que el sprint se logró satisfactoriamente según los PBI establecidos para este sprint. En este sprint, incluimos dos historias de usuarios a completar, PROC-1 Como cliente quiero un desarrollo de la arquitectura de la solución y PROC-7 Como cliente quiero un documento del proyecto.



Ilustración : Imagen PBI del Sprint 1

## Reunión de retrospectiva

Como ya he mencionado anteriormente, hemos utilizado Jira como herramienta de gestión de proyectos, pero cada semana dentro del sprint, hemos modificado la estimación de los PBI en el Sprint Backlog y casi siempre dicha reestimación ha sido al alza.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Ilustración : Gráfica de Burndown del sprint 1

# Sprint 2: Usuarios

## Reunión de preparación del Sprint

En este sprint definimos que queríamos poder mantener el estado del usuario por medio de las cookies, permitir el inicio de sesión mediante autenticación de terceros, en nuestro caso Google, y permitir la persistencia de nuestros usuarios gracias a MongoDB.

## Desarrollo del Sprint

Al comenzar este sprint, vimos que hay diferentes formas de mantener el estado del usuario, como las Cookies, el Local Storage o el Session Storage, cada uno con diferentes características, la más destacable, la capacidad. En la práctica vimos como implementar Local Storage, pero luego lo modificamos para adaptarlo a las cookies.

Después de conseguir mantener el estado, nos pasamos a implementar la autenticación de terceros de Google. Para ello, utilizamos Passport.js que es un middleware para implementar accesos OAuth de diferentes proveedores. Creamos un ID de cliente de OAuth desde GCP.

Tras tener el OAuth de Google, implementé el OAuth de GitHub y creé dos OAuth App en GitHub, una para producción y otras para las pruebas en local. Esta implementación fue bastante similar a la autenticación con Google, ya que Passport permite muchísimos tipos de proveedores y es muy sencillo incorporar nuevos proveedores.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Ilustración : Autenticación con GitHub

Una vez hecha la autenticación de terceros, implementamos la capa de acceso a datos y empezamos configurando la BBDD en Mongo Atlas y la conexión a ésta. También implementamos el Google One Tap para acceder al sistema mediante un modal para poder hacer peticiones a la API de autenticación de Google.

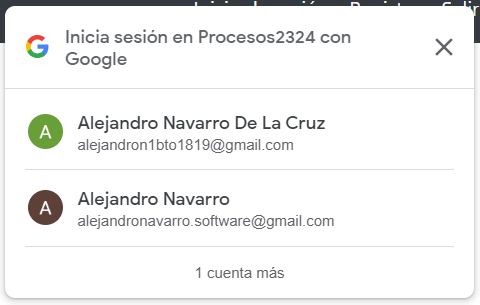


Ilustración : Modal de Google One Tap

Para terminar, implementé el registro e inicio de sesión para usuarios locales y dicha gestión con la base de datos anteriormente creada.

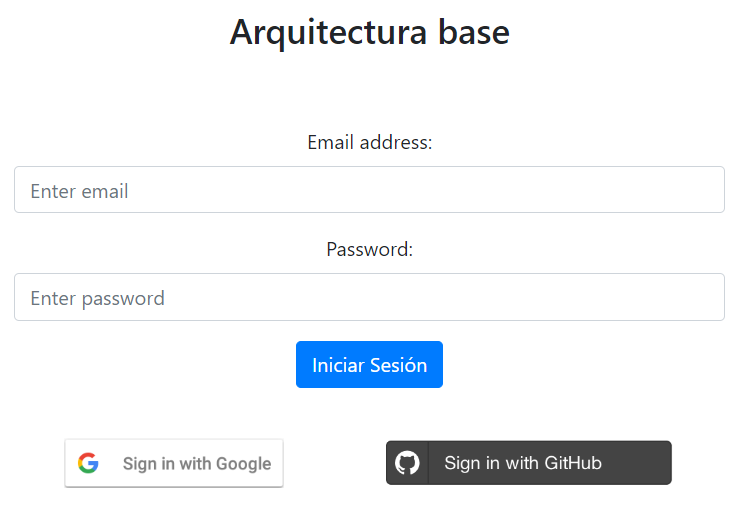


Ilustración : Diseño de Inicio de sesión

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración : Diseño de registro de usuarios

Además, para la elaboración de este sprint hemos cambiado el diagrama de la arquitectura de nuestra aplicación, añadiendo el componente de autenticación de Google, el middleware Passport, y la colección usuarios dentro de nuestra base de datos de MongoDB.

Diagrama

Descripción generada automáticamente

Ilustración : Diagrama de arquitectura sprint 2

## Reunión de revisión

Tras la revisión con el PO

## Reunión de retrospectiva

Hemos utilizado la herramienta de gestión de proyectos Jira,y hemos modificado la estimación de los PBI en el Sprint Backlog y casi siempre dicha reestimación ha sido al alza. Cabe destacar que la estimación de puntos historia en este Sprint, en mi caso no ha sido el acertado, ya que he terminado todos antes de la fecha de finalización del sprint.

Gráfico

Descripción generada automáticamente

Ilustración : Gráfica de Burndown del sprint 2

# Anexos

## Diagrama de secuencia: Agregar Usuario

En este diagrama de secuencia se muestran los pasos que ocurren en el sistema cuando le llega una petición para agregar un Usuario al sistema. Como vemos, cuando llega al sistema una petición, controlamos el evento de click en el elemento de visualización desde controlWeb.js, y creamos una instancia del clienteRest que se comunica con la capa de presentación de nuestro servidor y ésta con nuestra lógica, ubicada en modelo.js. Actualmente, como no es una solución definitiva y el registro de usuarios cambie, no he visto oportuno representar todas las alternativas de la lógica.

Texto

Descripción generada automáticamente con confianza media

Ilustración : Diagrama de secuencia AgregarUsuario

## Diagrama de secuencia: Inicio de Sesión en Local

En este diagrama se muestran los pasos para iniciar sesión con un usuario en local. En primer lugar, desde el iniciosesion.html, se muestra el formulario de inicio de sesión donde puedes ingresar el email y la contraseña para acceder. Una vez pulsado en el botón de Iniciar Sesión, dicho evento lo registra el controlWeb.js y envía al método iniciarSesion del clienteRest, el cual realiza un evento de POST al lado del servidor. El POST en el index.js llama al modelo.js y el modelo al cad.js y es ahí donde accede a la base de datos devolviendo el usuario si existe y un error en caso contario.

Diagrama

Descripción generada automáticamente

Ilustración : Diagrama de secuencia InicioSesionLocal