



**UNIVERSIDAD DE BUENOS AIRES**  
**FACULTAD DE INGENIERÍA**

Año 2020 - 1<sup>er</sup> Cuatrimestre

**Taller de Programación II (75.52)**

**ChoTuve**

**Manual de Proyecto**

Fecha de entrega: 30/07/2020

*Grupo 8*

79979 – Gonzalez, Juan Manuel ([juanmg0511@gmail.com](mailto:juanmg0511@gmail.com))

82449 – Laghi, Guido ([guido321@gmail.com](mailto:guido321@gmail.com))

86429 – Casal, Romina ([casal.romina@gmail.com](mailto:casal.romina@gmail.com))

96453 – Ripari, Sebastian ([sebastiandripari@gmail.com](mailto:sebastiandripari@gmail.com))

97839 – Daneri, Alejandro ([alejandrodaneri07@gmail.com](mailto:alejandrodaneri07@gmail.com))

# Contenido

<b>1. Objetivo</b>	<b>3</b>
<b>2. Equipo</b>	<b>4</b>
2.1 Miembros	4
2.2 Asignación de tareas	4
2.3 Aspectos de organización	5
<b>3. Gestión de proyecto</b>	<b>6</b>
3.1 Seguimiento de avance	6
3.2 Testing y seguimiento de issues	6
3.3 Diseño de la solución	6
3.4 Documentación	7
<b>4. Checkpoints</b>	<b>8</b>
4.1 Preparación de los checkpoints	8
4.2 Funcionalidad implementada por checkpoint	8
4.3 Cambio de alcance	10

# 1. Objetivo

El objetivo de este documento es presentar los aspectos macro de gestión y organización del proyecto, como por ejemplo la composición del equipo de trabajo, su organización, el seguimiento de avance, tratamiento de issues, diseño de features y la preparación de los distintos checkpoints, entre otros.

## 2. Equipo

En esta sección se describen los aspectos relativos al equipo de trabajo y su organización.

### 2.1 Miembros

Los miembros del equipo del proyecto son:

1. 79979 – Gonzalez, Juan Manuel ([juanmg0511@gmail.com](mailto:juanmg0511@gmail.com))
2. 82449 – Laghi, Guido ([guido321@gmail.com](mailto:guido321@gmail.com))
3. 86429 – Casal, Romina ([casal.romina@gmail.com](mailto:casal.romina@gmail.com))
4. 96453 – Ripari, Sebastian ([sebastiandripari@gmail.com](mailto:sebastiandripari@gmail.com))
5. 97839 – Daneri, Alejandro ([alejandrodaneri07@gmail.com](mailto:alejandrodaneri07@gmail.com))

Alumnos de la carrera de Ingeniería en Informática en la Facultad de Ingeniería de la Universidad de Buenos Aires.

### 2.2 Asignación de tareas

La asignación de tareas se realizó en dos capacidades:

1. Una asignación primaria, o principal, correspondiente a un componente de la solución por integrante. Para realizar esta asignación, en una primera reunión, cada integrante contó su experiencia previa y conocimientos, y se asignaron los componentes más afines a cada miembro. Esta distribución fue:

**AppServer** - *Romina Casal*

**MediaServer** - *Alejandro Daneri*

**AuthServer** - *Juan Manuel Gonzalez*

**WebAdmin** - *Sebastian Ripari*

**ClienteMobile** - *Guido Laghi*

El concepto detrás de esta distribución fue maximizar la capacidad de desarrollo de cada miembro del equipo, así como también la calidad de la solución.

2. Una asignación secundaria, variable por checkpoint o necesidad. A medida que avanzaban los desarrollos de cada miembro, se les asignaban otros componentes, para actuar en forma de soporte o ayuda al miembro principal.

La idea de esta asignación fue no desperdiciar horas hombre de desarrollo disponibles, ya sea porque un integrante no podía avanzar con su propio componente, o bien porque dicho componente se encontraba finalizado. Las asignaciones secundarias fueron:

Juan Manuel Gonzalez - **ClienteMobile**

Alejandro Daneri - **WebAdmin**

SebastiánRipari - **ClienteMobile**

Los recursos asignados con rol principal en el **ClienteMobile** y **AppServer** no tuvieron asignaciones secundarias, dado que estos componentes necesitaron de la mayor cantidad de esfuerzo y su desarrollo continuó hasta el checkpoint 4.

## 2.3 Aspectos de organización

El proyecto se organizó de la siguiente forma:

1. Un repositorio de GitHub principal:

<https://github.com/juanmg0511/fiuba-taller-2-documents>

Este repositorio contiene el *Kanban board* utilizado para la asignación y seguimiento de tareas, así como también un sitio hosteado en GitHub pages con la documentación del proyecto.

2. Un repositorio de GitHub para cada componente de la solución:

<https://github.com/romicasal/fiuba-taller-2-app-server>

<https://github.com/juanmg0511/fiuba-taller-2-auth-server>

<https://github.com/AlejandroDaneri/fiuba-taller-2-media-server>

<https://github.com/encubos/fiuba-taller-2-cliente-mobile>

<https://github.com/sebastianripari/fiuba-taller-2-web-admin>

El owner de cada repositorio es el recurso asignado a ese componente inicialmente. Se creó asimismo una *organization* con todos los miembros del equipo, a fin garantizar el acceso a los repositorios.

Los repositorios con código fuente poseen dos branches principales **master** y **develop**. Durante la etapa de desarrollo, los pushes se hicieron siempre a **develop**, mientras que luego de cada checkpoint se realizaba el merge a **master**. Al tener nuestro CI mapeado como **master:producción** y **develop:staging**, esto lograba tener en el ambiente productivo la versión del último checkpoint.

La colaboración dentro de un mismo repositorio se manejó por lo general mediante el uso de *pull requests*.

Se puede encontrar más información sobre los repositorios en los manuales de diseño de cada uno de los componentes de la solución.

## 3. Gestión de proyecto

En esta sección se exponen aspectos principales referentes a la gestión del proyecto.

### 3.1 Seguimiento de avance

En cuanto al seguimiento del avance, se trabajó de la siguiente manera:

1. Todas las tareas del proyecto fueron cargadas en un *Kanban board* donde se definieron los cuatro checkpoints, y los componentes de la solución.
2. El tiempo de cada checkpoint se dividió en sprints semanales, con una reunión fija los días viernes por la tarde para revisar el avance y programar las tareas de la semana siguiente. Estas reuniones fueron organizadas en forma remota mediante Google Meet.

El tablero se implementó en GitHub, y puede consultarse en la URL:

<https://github.com/juanmg0511/fiuba-taller-2-documents/projects/7>

### 3.2 Testing y seguimiento de issues

Para el seguimiento de issues y tareas menores, se utilizó una planilla de Google Docs, con una pestaña por componente, donde se crearon tablas con **ID**, **Descripción**, **Recurso asignado** y **Estado**, a fin de llevar un seguimiento detallado del trabajo realizado.

Esta planilla fue revisada en las reuniones mencionadas en la sección anterior. Puede consultarse en:

[https://docs.google.com/spreadsheets/d/1KdFne3taoJg1qzLA5YFWb-iGzZXnoHoF8ba0peiMMfY/edit?usp=drive\\_web&oid=108812359868862507059](https://docs.google.com/spreadsheets/d/1KdFne3taoJg1qzLA5YFWb-iGzZXnoHoF8ba0peiMMfY/edit?usp=drive_web&oid=108812359868862507059)

### 3.3 Diseño de la solución

En cuanto a las decisiones de diseño de la solución, se trabajó de la siguiente manera:

1. En primera instancia, en reuniones de equipo específicas se esbozó la arquitectura macro de la solución, y se diseñaron las API de los servers. También se definió la tecnología a utilizar para cada uno de los componentes de la solución.
2. Los detalles de implementación y estrategia de desarrollo fueron dejados a criterio de cada integrante.

3. Conforme fue avanzando el desarrollo, se tomaban nuevas decisiones críticas en conjunto, aprovechando las reuniones de seguimiento.

Para la comunicación diaria, así como también para consultas generales se utilizó activamente el workspace de Slack proporcionado por la cátedra, en un canal privado del equipo.

### 3.4 Documentación

La documentación del proyecto fue realizada utilizando las siguientes herramientas:

1. Swagger.io para documentar las API de los servers.
2. Draw.io para realizar los diagramas.
3. Google Docs para los manuales.
4. GitHub pages para la presentación y hosting.

La página donde se puede consultar la documentación es:

<https://juanmg0511.github.io/fiuba-taller-2-documents/>

## 4. Checkpoints

En esta sección se describen las cuestiones pertinentes a los checkpoints.

### 4.1 Preparación de los checkpoints

Previo a la reunión con los docentes para realizar el checkpoint, se hizo una reunión de equipo donde se coordinó la presentación a dar, así como también ultimar detalles como por ejemplo limpieza de datos o creación de usuarios.

### 4.2 Funcionalidad implementada por checkpoint

A continuación se detallan las funcionalidades implementadas a nivel macro, para cada uno de los checkpoint.

Como estrategia fundamental, se apuntó como primer medida a provisionar y configurar todos los aspectos de infraestructura, a fin de poder tener todos estos elementos disponibles y funcionando.

Por ejemplo, para un server, “configuración inicial” implica:

1. Configuración de ambiente local con Docker/Compose
2. Creación de branches en GitHub
3. Creación de apps en Heroku, instalación y configuración de add-ins
4. Configuración de CI en Travis
5. Integración de tests al proyecto al momento de deploy.

Sólo luego de tener estos aspectos funcionando (con contenido básico, o stubs/placeholders) se procedió a desarrollar la funcionalidad requerida.

En las tablas expuestas a continuación se mencionan las funcionalidades macro implementadas por cada checkpoint. La funcionalidad figura en el checkpoint en que se disponibilizó por primera vez, siendo refinada en caso de requerirse *a posteriori*:

	<b>AppServer</b>
<b>Checkpoint 1</b>	Configuración inicial Home Enpoint de ping Endpoint de status
<b>Checkpoint 2</b>	API de AuthServer Endpoint de videos
<b>Checkpoint 3</b>	Endpoint de comentarios Endpoint de amistades



<b>Checkpoint 4</b>	Reglas para muro Endpoint de stats Documentación
---------------------	--

	<b>AuthServer</b>
<b>Checkpoint 1</b>	Configuración inicial Home Endpoint de usuarios Endpoint de ping Endpoint de status
<b>Checkpoint 2</b>	Endpoint de adminusers Endpoint de sessions Endpoint de recovery Endpoint de stats Endpoint de requestlog Documentación
<b>Checkpoint 3</b>	-
<b>Checkpoint 4</b>	-

	<b>MediaServer</b>
<b>Checkpoint 1</b>	Configuración Inicial Home Endpoint de ping Endpoint de status Endpoint de videos
<b>Checkpoint 2</b>	Endpoint de avatars
<b>Checkpoint 3</b>	Documentación
<b>Checkpoint 4</b>	-

	<b>WebAdmin</b>
<b>Checkpoint 1</b>	Configuración inicial Sección de videos Sección de usuarios Sección de administradores Sección de status
<b>Checkpoint 2</b>	Funcionalidad de recovery
<b>Checkpoint 3</b>	-
<b>Checkpoint 4</b>	Estadísticas / gráficos

	<b>ClienteMobile</b>
<b>Checkpoint 1</b>	Login Home Subir Video
<b>Checkpoint 2</b>	Reproducción de video Perfil Muro
<b>Checkpoint 3</b>	Comentarios de videos Avatars Sign in with Google Logout
<b>Checkpoint 4</b>	Compatibilidad iOS Amistades Documentación

### 4.3 Cambio de alcance

Luego del tercer checkpoint, se acordó con el docente asignado al grupo un cambio de alcance al enunciado original: teniendo como objetivo generar una App de mayor calidad final y más terminada en lo que refiere a los detalles, se propuso no implementar la funcionalidad de *chat* pedida en el enunciado, visto el tiempo de proyecto restante, para concentrar el esfuerzo en pulir la funcionalidad ya implementada.

Asimismo, también se propuso brindar soporte para dispositivos iOS, algo no contemplado en los requerimientos, pero que a criterio del grupo valía la pena implementar, dado el uso de React Native como tecnología para el **ClienteMobile**.

Cabe aclarar que para poder lograr esto último se necesita de cierto esfuerzo, ya que hay que adaptar partes de código para que la solución funcione adecuadamente en iOS (sign in with Google, permisos, push notifications).