

## Technical Report

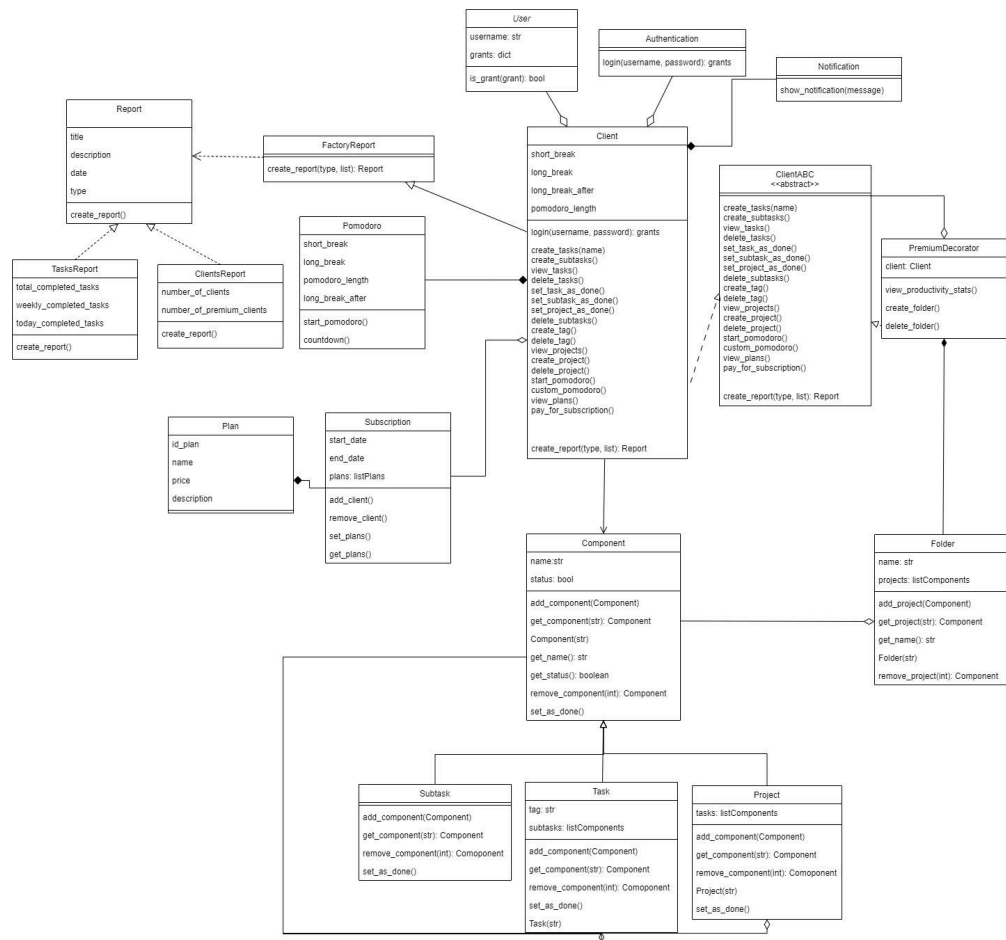
- Andres Jose Acevedo Avila
- Javier Camilo Murcia Nahan

### Diagrams:

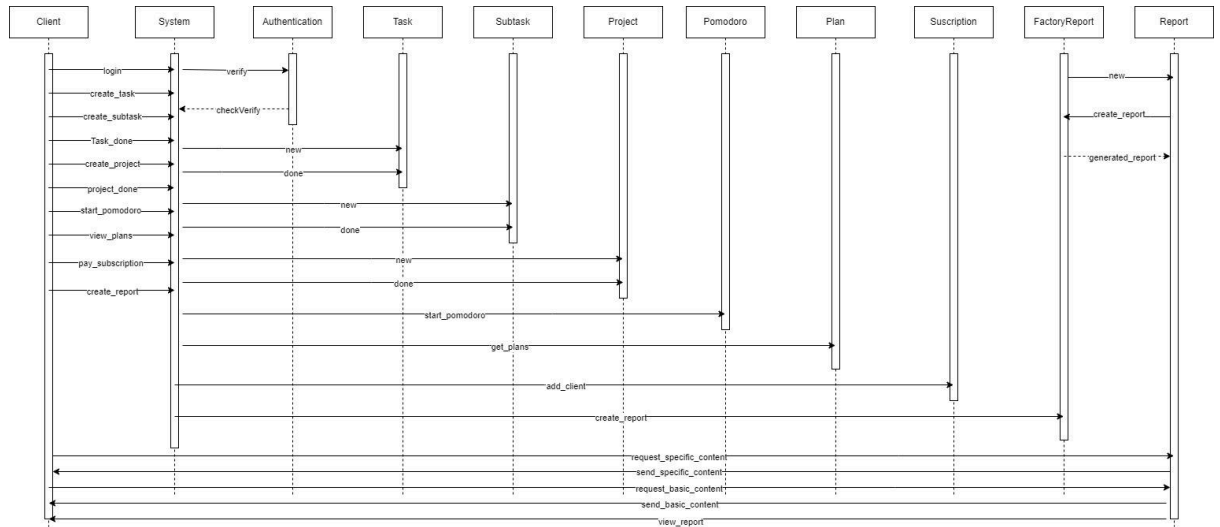
#### - CRC Cards

<table><tr><th colspan="2">User</th></tr><tr><td>- Verify what the customer can access</td><td>Client</td></tr></table>	User		- Verify what the customer can access	Client	<table><tr><th colspan="2">Authetication</th></tr><tr><td>- Log in - Log out</td><td>Client</td></tr></table>	Authetication		- Log in - Log out	Client	<table><tr><th colspan="2">SubTask</th></tr><tr><td>-Store and manage subtasks -Add, get, and remove subtask</td><td>Component</td></tr></table>	SubTask		-Store and manage subtasks -Add, get, and remove subtask	Component	<table><tr><th colspan="2">Notification</th></tr><tr><td>- Send and show notification -Store the message</td><td>Client</td></tr></table>	Notification		- Send and show notification -Store the message	Client								
User																											
- Verify what the customer can access	Client																										
Authetication																											
- Log in - Log out	Client																										
SubTask																											
-Store and manage subtasks -Add, get, and remove subtask	Component																										
Notification																											
- Send and show notification -Store the message	Client																										
<table><tr><th colspan="2">Client</th></tr><tr><td>- Create, edit, view and delete tasks - Create, edit, view and delete subtasks - Create, view and delete tag - Create, edit, view and delete project - Receive notifications - Start, stop and custom pomodoro timer - See plans - Pay for subscription - Get and create report premium clients -Get and create report clients</td><td>User Authentication Task Subtask Tag Project Notification Pomodoro Plan Subscription</td></tr></table>	Client		- Create, edit, view and delete tasks - Create, edit, view and delete subtasks - Create, view and delete tag - Create, edit, view and delete project - Receive notifications - Start, stop and custom pomodoro timer - See plans - Pay for subscription - Get and create report premium clients -Get and create report clients	User Authentication Task Subtask Tag Project Notification Pomodoro Plan Subscription	<table><tr><th colspan="2">Component</th></tr><tr><td>- Add, obtain and remove components - Store the name and status of the component - Mark the component as done</td><td>Subtask Task Project Folder Client</td></tr></table> <table><tr><th colspan="2">Folder</th></tr><tr><td>- Store folder name -Manage projects -Add, get, and remove projects</td><td>Component Client</td></tr></table>	Component		- Add, obtain and remove components - Store the name and status of the component - Mark the component as done	Subtask Task Project Folder Client	Folder		- Store folder name -Manage projects -Add, get, and remove projects	Component Client	<table><tr><th colspan="2">Task</th></tr><tr><td>-Store and manage tasks -Add, get, and remove task -Mark the task as done</td><td>Component</td></tr></table> <table><tr><th colspan="2">Projects</th></tr><tr><td>-Store and manage projects -Add, get, and remove projects -Mark the project as done</td><td>Component</td></tr></table>	Task		-Store and manage tasks -Add, get, and remove task -Mark the task as done	Component	Projects		-Store and manage projects -Add, get, and remove projects -Mark the project as done	Component	<table><tr><th colspan="2">Pomodoro</th></tr><tr><td>- Store short and long break duration - Store pomodoro length -Store number of Pomodoros before a long break</td><td>Client</td></tr></table>	Pomodoro		- Store short and long break duration - Store pomodoro length -Store number of Pomodoros before a long break	Client
Client																											
- Create, edit, view and delete tasks - Create, edit, view and delete subtasks - Create, view and delete tag - Create, edit, view and delete project - Receive notifications - Start, stop and custom pomodoro timer - See plans - Pay for subscription - Get and create report premium clients -Get and create report clients	User Authentication Task Subtask Tag Project Notification Pomodoro Plan Subscription																										
Component																											
- Add, obtain and remove components - Store the name and status of the component - Mark the component as done	Subtask Task Project Folder Client																										
Folder																											
- Store folder name -Manage projects -Add, get, and remove projects	Component Client																										
Task																											
-Store and manage tasks -Add, get, and remove task -Mark the task as done	Component																										
Projects																											
-Store and manage projects -Add, get, and remove projects -Mark the project as done	Component																										
Pomodoro																											
- Store short and long break duration - Store pomodoro length -Store number of Pomodoros before a long break	Client																										
<table><tr><th colspan="2">Plan</th></tr><tr><td>-Store plan name, price and description</td><td>Client</td></tr></table>	Plan		-Store plan name, price and description	Client	<table><tr><th colspan="2">Report</th></tr><tr><td>-Manage basic report information: title, description, date, and type -Provide methods to fill in basic information -Create the report</td><td>Client</td></tr></table>	Report		-Manage basic report information: title, description, date, and type -Provide methods to fill in basic information -Create the report	Client	<table><tr><th colspan="2">TasksReport</th></tr><tr><td>- Manage specific task information such as total completed tasks, weekly completed tasks, today completed tasks, task focus time, and project time</td><td>Report</td></tr></table>	TasksReport		- Manage specific task information such as total completed tasks, weekly completed tasks, today completed tasks, task focus time, and project time	Report	<table><tr><th colspan="2">ClientsReport</th></tr><tr><td>- Manage specific client information such as the number of clients and the number of premium clients - Specific report</td><td>Report</td></tr></table>	ClientsReport		- Manage specific client information such as the number of clients and the number of premium clients - Specific report	Report								
Plan																											
-Store plan name, price and description	Client																										
Report																											
-Manage basic report information: title, description, date, and type -Provide methods to fill in basic information -Create the report	Client																										
TasksReport																											
- Manage specific task information such as total completed tasks, weekly completed tasks, today completed tasks, task focus time, and project time	Report																										
ClientsReport																											
- Manage specific client information such as the number of clients and the number of premium clients - Specific report	Report																										
<table><tr><th colspan="2">Subscription</th></tr><tr><td>-Store start date and finish date -Add and remove a client -Modify plan</td><td>Client Plan</td></tr></table>	Subscription		-Store start date and finish date -Add and remove a client -Modify plan	Client Plan																							
Subscription																											
-Store start date and finish date -Add and remove a client -Modify plan	Client Plan																										

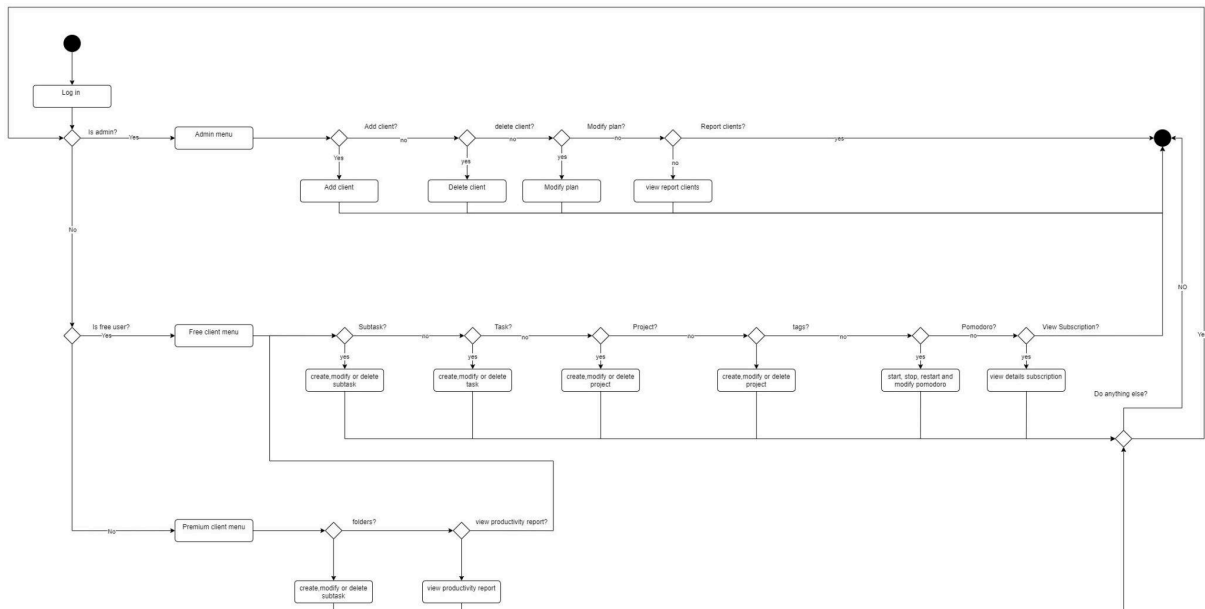
- **Class Diagram**



## - Sequence Diagram



## - Activity Diagram:



## Decisiones Técnicas:

As end users, only one client was defined, and it was planned to use authentication permissions to designate the admin, as they only had one function: generating client reports.

The template pattern was initially considered for generating the report, since there was a static part (general report data) and a dynamic part depending on the type of object (specific report data). However, it was realized that the report follows the same structure with different content depending on the report type, so the factory pattern was used to allow report generation from the child classes.

The composite pattern was utilized for the creation of subtasks, tasks, and projects. The creation of folders was not considered within the composite, as they are not contained by

anything. The decorator pattern was planned to define additional functions for premium clients.

The proxy pattern could have been used to handle the logging of completed tasks as a cache, but it would have added unnecessary complexity to the program, given that simply sending a list of completed tasks was sufficient.

As design patterns were utilized, the SOLID principles are reflected, and spaghetti code was avoided, as each class had its own responsibility. With the factory pattern, we see the Liskov substitution principle in action, and the main implementation is improved, allowing for openness to extension and closure to modification.

### **Consideraciones:**

This version is incomplete due to poor time management by the developers. Although all the classes were written, not all of them were used, such as the notification class. Additionally, the concrete part of report generation, adding to the subscription, premium client functions, the implementation of web services, and the database are still missing. The developers learned that they need to better organize their time management and not take it for granted.