

Daily Scrum Meeting Minutes:

Attendees:

Bryan Garcia - 6289119

Tom Mizrahi - 6358028

Andres Jaimes - 6289563

Cesar Pereyra - 5544950

Bryan Hernandez - 6456561

Start time: 3:15 PM

End time: 6:15 PM

Bryan Garcia:

- **How many hours have I worked since the last meeting?**
 - 3 hours
- **What was done since the last scrum meeting?**
 - Finish off the final dailys
- **What is planned to be done until the next scrum meeting?**
 - Rewind and remember the practices we did
- **What are the hurdles?**
 - Present and win

Tom Mizrahi:

- **How many hours did I work since the last meeting?**
 - 3 hours
- **What was done since the last scrum meeting?**
 - Finish dailys
- **What is planned to be done until the next scrum meeting?**
 - Finish dailys
- **What are the hurdles?**
 - None yet

Cesar Pereyra:

- **How many hours have I worked since the last meeting?**
 - 3 hours
- **What was done since the last scrum meeting?**
 - Presented our project.
- **What is planned to be done until the next scrum meeting?**
 - **Sleep**
- **What are the hurdles?**
 - Receiving our grade!

Bryan Hernandez:

- **How many hours did I work since the last meeting?**

- 3 hours
- What was done since the last scrum meeting?
 - Continue progressing my knowledge on the softwares
- What is planned to be done until the next scrum meeting?
 - Strengthen the security of the project.
- What are the hurdles?
 - None right now

Andres Jaimes:

- **How many hours have I worked since the last meeting?**
 - 3
- **What was done since the last scrum meeting?**
 - created the MultiDNS fault tolerant cdn for routing the dns setup in aws using fastly and cloudflare
 - setup a decently strong ec2 instance that will play the role of the reverse proxy for the backend container
 - created a redirection Domain name with its own redirection page to make sure theres no weird transitions for the users
- **What is planned to be done until the next scrum meeting?**
 - finalize the route 53 setup
 - give ec2 a SSL to be encrypted fully in https
 - encrypt The S3 bucket so there's also encryption in data transferring as well
 - This would finalize PROTOTYPE #2
- **What are the hurdles?**
 - dns conflicts
 - awkward transitions between each cdn required multiple new ideas
 - had to prioritize time over perfection due to AWS not giving me access to a load balancer so i can have a full containerized communicated project working inside its own private subnet while the load balancer is part of the public subnet redirecting traffic on each end to integrate security