**Preliminary Implementation Outline**

The design of a walking laboratory agent involves analyzing a complex system that interacts with the environment, makes decisions and adapts its behavior according to conditions. This complex system performs locomotive activities, therefore, the agent must be able to analyze, learn and act on them. Considering the above, for the design it is necessary to be able to simulate this agent with locomotive functions and its behavior in the environment. The simulation of the agent is of great help for the design, and tools that help us with this are Gymnasium and Stable-Baselines3.

Gymnasium allows us to model the physical conditions of a walking agent, thanks to the compatibility with MuJoCo engines, which contains environments where humanoids are simulated with goals of moving forward until falling, allowing us to design bipedal agents adapted to the laboratory. Moreover, Stable-Baselines3 includes algorithms such as DQN, PPO, SAC, A2C, which serve for effective reinforcement learning in locomotion.

The gait lab, focused on advanced recommended diagnostics, injury prevention, and personalized diagnostics, could implement Q-Learning and DQN-related algorithms in the following order, from the most basic to the most advanced:

1. Classic Q-Learning (Tabular): It is the foundation to understand how reinforcement learning works. Classic Q-Learning: Learns a table that indicates the best action in each situation. Useful for small environments with few options.
2. DQN (Deep Q-Learning): Uses a neural network to decide what to do when there are many variables. Ideal when the data is complex, such as from biomechanical sensors.
3. Double DQN: Improves DQN so that it doesn't rely too much on its own predictions, making it more accurate in decision-making.
4. Dueling DQN: Separates the importance of the state and the action to make better decisions. Works well when there are many similar actions.
5. DQN with Prioritized Experience Replay (PER): Learns more about important or rare cases. Useful if there are difficult or special examples in the data.
6. A2C/A3C (Actor-Critic Advantage): Has two networks: one decides and the other evaluates whether it was a good decision. Good for tasks that are performed in steps, such as walking or running.

7. DDPG / TD3: Used when decisions aren't yes/no, but rather continuous adjustments (such as moving a joint). Ideal for precise body control.