To localize a robot and navigate it to a specific position without prior knowledge of the map, we would typically need some combination of the following sensors:

1. **Odometry or Wheel Encoders:** These sensors measure the rotation of the robot's wheels to estimate its position and track its movement. They can provide information about the distance traveled and the direction of movement.
2. **Inertial Measurement Unit (IMU):** An IMU consists of sensors such as accelerometers and gyroscopes, which can provide data on the robot's orientation, angular velocity, and acceleration. This information is useful for estimating the robot's position and detecting changes in direction.
3. **Distance Sensors:** These sensors, such as ultrasonic or infrared range finders, can measure the distance between the robot and nearby obstacles. They are useful for detecting obstacles in the environment and avoiding collisions.
4. **Vision Sensors:** Cameras or depth sensors, such as RGB-D cameras or LIDAR, can provide visual or depth information about the robot's surroundings. These sensors enable the robot to perceive its environment, detect objects, and estimate distances.
5. **Wireless Beacons or Localization Markers:** These are physical markers or beacons placed in the environment. The robot can use their known positions to estimate its own position relative to the markers. This technique is often used in techniques like landmark-based localization or simultaneous localization and mapping (SLAM).

Regarding the algorithm to implement, a common approach for robot localization and navigation in an unknown environment is the **Simultaneous Localization and Mapping (SLAM)** algorithm. SLAM enables the robot to build a map of its surroundings while simultaneously estimating its own position within that map.

SLAM typically combines sensor data from odometry, IMU, distance sensors, and vision sensors to create a map of the environment and localize the robot within that map. The robot moves through the environment, collecting sensor data and updating its map and position estimates in real-time.

There are various SLAM algorithms available, ranging from filtering-based techniques like Extended Kalman Filter (EKF) SLAM, to graph-based methods like GraphSLAM, to modern approaches such as FastSLAM or ORB-SLAM. Implementing a SLAM algorithm can be complex, and the choice of algorithm depends on factors like computational resources, sensor capabilities, and the specific requirements of the application. It is recommended to use existing SLAM libraries or frameworks, such as Google Cartographer, ROS (Robot Operating System) navigation stack, or OpenSLAM, which provide implementations of various SLAM algorithms and can be adapted to your specific robot platform and sensor setup.

By utilizing the appropriate sensors and implementing a SLAM algorithm, the robot can autonomously explore and navigate an unknown environment while localizing itself to reach a specific position.