**Técnicas agrícolas utilizadas en cultivos hidropónicos o Fase de Requerimientos**

* **Caracterización Guasca**
  + **Lugar**
  + **Siembra**
  + **Cultivo**
* **Requerimientos funcionales y no funcionales**
* **Análisis del estado del arte**

**Diseño de la red**

* **Estándares de comunicación (Elección/comparativa)**
* **Comparación de tecnologías de redes inalambricas**

Tabla 4. Comparación de características.

|  |  |  |  |
| --- | --- | --- | --- |
| **Características** | **Bluetooth** | **WLAN** | **Zigbee** |
| **Protocolo IEEE** | 802.15.1 | 802.11 g/g | 802.15.4 |
| **Rango en metros** | 10 | 50-100 | 10-100 |
| **Data rate (bit/s)** | 1M | 11 – 54M | 20, 40, 250 k |
| **Duración de batería (días)** | 7 | 0.5 – 5 | >100 |
| **Frecuencia de operación (Hz)** | 2.4 G | 2.4 – 5 G | 868M, 902-926M y 2.4G |
| **Complejidad** | Alta | Alta | Bajo |
| **Consumo de potencia (mW)** | 195 | 1050 | 72 |
| **Seguridad** | 64 – 128 kbits | IEEE 802.11i (WPA2) | 128 bit AES y seguridad de la capa de aplicación |
| **Aplicación** | Comunicación inalámbrica | Internet, sitios web, email | Monitorization y control |

Fuente: Propia.

GONZÁLEZ, Ricardo y VILLAPOL, María. Un estudio comparativo de las tecnologías para redes de área personal [en línea]. 2013. [Consultado: 27 de octubre de 2020]. Disponible en: https://www.researchgate.net/publication/266080992\_Un\_Estudio\_Comparativo\_de\_las\_Tecnologias\_para\_Redes\_de\_Area\_Personal

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tecnología** | **Topología** | **Tamaño de red** | **Tecnología de radio** | **Velocidad de transmisión** | **Banda de frecuencia** | **Alcance** |
| Wifi | -Punto a punto  -Punto a multipunto  -Multipunto a multipunto | 32 nodos | IEEE 802.11 b/g/ n | 5,9 Mbit/s TCP  7,1 Mbit/s UDP | 2,4 GHz | 30m |
| Lorawan | -Punto a punto  -Estrella | Hasta 1 millón de nodos | LoRa Alliance | 0.3 kbps - 50 kbps | 433 MHz, 868 MHz y 915 MHz | >20 km |
| Zigbee | -Malla | Hasta 65.000 nodos | IEEE 802.15.4 | 250 kbps | 2.4 GHz | Hasta 300 + m (línea de visión)  75-100 m interior |
| Bluetooth | -Punto a punto  -Punto a multipunto | 7 nodos | IEEE 802.15.1 | 721.2 kbps v1.2  2.1 Mbps v2.0 +EDR | 2.4 GHz | 10m |
| Sigfox | -Estrella | Hasta 250.000 nodos | Ultra Narrow Band (UNB) | 10-1000bps | 900 MHz | 30-50km (rural) y 3-10km (urbanos) |
| Thread | - | 250 nodos | Basado en IEEE 802.15.4 y 6LowPAN | - | 2,4 GHz | - |
| Neul | - | - | Basado en chip Iceni y Weightless | 100kbps | 900MHz, 458MHz, 470-790MHz | 10km |
| Z-Wave | -Punto a punto  -Malla | 232 nodos | Z-Wave Alliance ZAD12837 / ITU-T G.9959 | 9,6/40/100 kbps | 900MHz | 30m |

* **Hardware (Elección/comparativa)** 
  + **Módulos, sensores, SBC (single board computer)**
* **Características Xbee 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Especificaciones técnicas del Xbee 3 Zigbee 3.0** | | | |
| **Rendimiento** | | **Especificaciones Generales** | |
| **Rango de interiores/urbano** | 60 metros | **Frecuencia de operación** | ISM 2.4 GHz |
| **Rango con línea de vista** | 1200 metros | **Dimensiones** | 2.438 x 2.761 cm |
| **Potencia de Tx** | 8 dBm | **Temperatura de operación** | -40º C a 85º C |
| **Sensibilidad de Rx** | -103 dBm | **Opciones de antena** | PCB Antenna, U. FL Connector, RPSMA Connector |
| **Velocidad de Tx** | 250 Kbps | **Memoria** | MB / 128 KB RAM (32KB disponibles para MicroPython) |
| **Inmunidad a Interferencia** | DSSS (espectro ensanchado por frecuencia directa) | **Métodos de configuración** | API, comandos AT, local u OTA |
| **Requerimientos de potencia** | | **Seguridad y red** | |
| **Voltaje de alimentación** | 2.1 a 3.6 V | **Opciones de dirección** | PAN ID y Direcciones de clúster y puntos finales (opcional) |
| **Corriente de Tx** | 40 mA @ 8 dBm | **Protocolos** | Zigbee 3.0 |
| **Corriente de Rx** | 17 mA | **Canales** | 16 |
| **Corriente de apagado** | 2 micro Amp @ 25º C (77º F) | **Encriptación** | 128/256 bit AES |

* **Topología de red (Elección/comparativa y ubicación en sitio)**

5.2.8 Topología de redes Xbee. Los módulos Xbee soportan múltiples protocolos inalámbricos, apropiados para las diferentes topologías de red. Los estándares abiertos incluyen Zigbee, 802.15.4 y WiFi. El fabricante Digi, ha desarrollado sus propios protocolos como Multipunto y DigiMesh[[1]](#footnote-1). A continuación, se presentan las topologías de red:

Tabla 5. Topología de redes Xbee

|  |  |  |
| --- | --- | --- |
| Estándar | Topología | Uso |
| 802.15.4 | Punto a punto | * Un solo punto de comunicación * Rápida comunicación entre dispositivos |
| Punto a multipunto | * Redes no expandibles que necesitan poca energía o un funcionamiento intermitente |
| Zigbee/Zigbee SE |  | * Sistemas grandes que necesitan expandirse sin una pérdida de función. * Sistemas que necesitan comunicaciones extendidas. * Sistemas que utilizan patrones de comunicaciones no direccionales. * Sistemas con función intermitente de los módulos individuales debido a pérdida de potencia o funcionamiento cíclico. * Redes a gran escala con bajo consumo de energía en los dispositivos finales. * Sistemas que requieren interoperabilidad entre dispositivos fabricados por diferentes proveedores. |
| DigiMesh |  | * Sistemas que requieren el modo sleep en todos los nodos. * Sistemas que requieren simplificación, configuración y expansión de la red. * Redes de malla más robustas (no dependencias padres / hijo). * Entornos donde una mayor confiabilidad es importante debido a los enrutadores que van y vienen debido a interferencias o daños. |
| Multipunto |  | * Redes multipunto con opciones de mayor alcance. * Sistemas que no requieren una comunicación rápida |

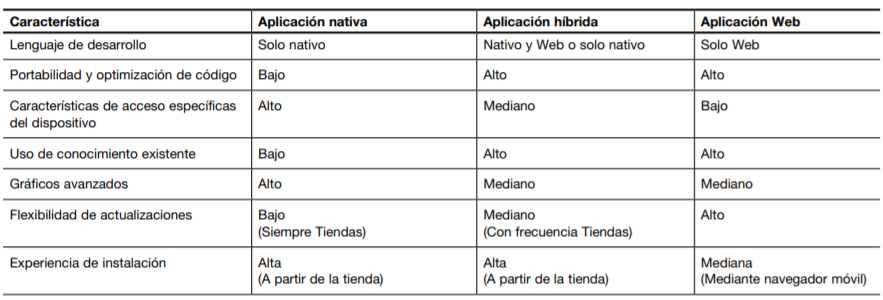
Fuente: Ibid.

* **Roles de los Nodos**
* **Captura de informacion**
* **Esquema de monitorización de la red**
  + **Diseño del sistema de gestión de nodos**
  + **Hardware**
  + **Software**
  + **Gestión de Fallas**
  + **Pruebas piloto**

**Implementación del sistema de monitoreo**

* **Instalación de nodos en área de cultivo**
* **Pruebas piloto de funcionamiento ¿**

**Diseño de la aplicación**

* **Tipos de aplicaciones (Elección/comparativa)**
* Tabla 1. Enfoques de aplicaciones móviles.
* 
* Fuente: IBM. Op. Cit., p. 8.
* **Lenguajes de programación/ (Elección/comparativa)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | PHP | Perl | Ruby | JavaScript | Python |
| Estructura | Orientado a objetos | Lenguaje de programación de bajo nivel / Orientado a objetos | Lenguaje de scripts y orientado a objetos. | Lenguaje de scripts y basado en objetos. | Orientado a objetos. |
| Licencia | Open Source, pero con restricción sobre el nombre. | Open Source | Open Source | Gratuito, pero no Open Source. | Open Source sin necesidad de entregar su código fuente. |
| Bases de datos | Oracle, Sybase, PostgreSQL, InterBase, MySQL, SQLite, MSSQL, Firebird, entre otras. | Oracle, Sybase, Postgres, MySQL, MongoDB, SQLite | SQL, Oracle, MongoDB, MariaDB, MySQL, SQLite, FireBird, entre otros. | SQLite, WebSQL, IdexedDB, PostgreSQL, MySQL, MongoDB, MariaDB, entre otros. | Oracle, PostgreSQL, MySQL, SQLite, |
| Sistemas Operativos | Windows, Linux, FreeBSD, Mac, Novell, Netware, RISC Los, AIX, IRIX y Solaris | Windows, Mac, Linux, Solaris, BSD, entre otros. | Linux, Unix, Mac, Windows, DOS, BeOS, OS/2, entre otros. | Su versión para servidores Node.js es soportada en: macOS, Linux, smartOS, FreeBSD, Windows. | Aquellos que soporten Apache 2 con mod\_python. |
| Tipado | Tipado dinámico | Tipado dinámico | Tipado dinámico | Tipado dinámico | Tipado dinámico |
| Procesamiento | Se lleva a cabo en el servidor. | Se lleva a cabo en el servidor, con ayuda de un intérprete. | Se lleva a cabo en el servidor con un interprete de ruby. | Se lleva a cabo en el equipo del cliente, pero también puede ejecutarse en el servidor, dependiendo de su versión. | Se lleva a cabo en el servidor, pero requiere un intérprete. |
| Protocolos | IMAP, SNMP, NNTP, POP3, HTTP, LDAP, XML-RPC, SOAP, entre otros. | HTTP, TCP/IP, SOAP Lite, IMAP, XML-RPC, entre otros. | SOAP, XML-RPC, HTTP, TCP/IP, IMAP. | HTTP, TCP/IP, XML-RPC, IMAP, entre otros. | HTTP, TCP/IP, XML-RPC, IMAP, SNMP, LADP, SOAP, entre otros. |
| Características especiales | Código portable,  Veloz y robusto, es un lenguaje de programación de dominio especifico. | Lenguaje interpretado, no tiene control de excepciones | Garbage collector, lenguaje interpretado, la comunidad y el soporte no es tan extenso, su procesamiento es más lento. | Se encuentra contenido dentro de un documento HTML, lenguaje interpretado, dinámico, multiplataforma, entre otras. | Lenguaje interpretado, multiplataforma, posee garbage collector, |
| Aplicaciones |  | Administración de sistemas operativos, creación de formularios WEB | Desarrollo web, desarrollo multiplataforma, | Desarrollo de aplicaciones cliente-servidor y aplicaciones similares a los CGI, | Inteligencia artificial, Data Science, Big Data, Desarrollo web, |

* **Frameworks**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Angular.Js | React.Js | Vue.Js | Ember.Js | Meteor.Js | BlackBone.Js |
| Plataforma | Multiplataforma | Web | Web | Web | Multiplataforma | Web |
| Patrón de diseño | Model View Controller (MVC) | responsable solo de la capa de visualización. | Model View Controller (MVC) | Model View Controller (MVC) |  | Model View Controller (MVC) |
| Licencia | Open Source | Open Source | Open Source | Open Source | Open Source | Open Source |
| Data Binding | Si | Usa en equivalente conocido como DOM Virtual. | Si | Si | Si | Si |
| Backend | Java, Python, Ruby, c#, entre otros |  | Python, PHP, Ruby, entre otros | Java, Ruby, Python, entre otros. | PHP, Ruby,.Net, Python, entre otros. | Java, Python, Ruby, c#, entre otros |
| Comunicación al servidor | Peticiones Http | Peticiones Http | Peticiones Http | Peticiones Http | Peticiones Http y DDF | Peticiones Http desde función interna Backbone.sync. |
| Características | Ligero, buena gestión de dependencias, desacoplamiento del DOM, entre otras. | Utiliza JSX, posee isomorfismo, que permite renderizar mismo código para cliente y servidor | Gran velocidad, fácil de integrar | La curva de aprendizaje es muy pronunciada, actualiza el DOM asíncronamente. | Mantiene conexión de bajo nivel con el servidor por medio de DDF. Esta soportado sobre Node.js. Mismo lenguaje para el cliente y el servidor. | Es un framework mínimo con menos de 800 líneas de código. |
| Documentación | Posee gran cantidad de documentación | Posee gran cantidad de documentación | La documentación es abundante. | No posee mucha documentación desde el cambio de la versión 2.1.0 | Posee una gran cantidad de documentación y recursos. | Posee una gran cantidad de documentación. |
| Complejidad | Media | Media | Baja | Alta |  | Alta |
| Aplicaciones | Aplicaciones Web basadas en Javascript del lado del cliente. | Aplicaciones web completas del lado del cliente. | Puede desarrollar aplicaciones web y móviles, dependiendo de su versión. | Aplicaciones web ambisiosas. | Aplicaciones móviles y aplicaciones web. | Aplicaciones SPA (Single Page Application) |

* **Motores de bases de datos (Elección/comparativa)**
* **Servicios en la nube (almacenamiento y procesamiento)**
* **Inteligencia artificial**
* **Diagramas UML**
* **Desarrollo de aplicación**
* **Backend**
* **Frontend**

**Verificación y pruebas del prototipo**

* **Pruebas de funcionamiento de hardware**
* **Pruebas de funcionamiento de software**
* **Capacitación a los agricultores**
* **Limitaciones**
* Resultado de las pruebas

1. DIGI. Digi XBee® Zigbee® Mesh kit [en línea]. Digi International Inc. 2019. p. 191-192. [Consultado: 25 de octubre de 2020]. Disponible en: https://www.digi.com/resources/documentation/digidocs/pdfs/90001942-13.pdf [↑](#footnote-ref-1)