

Veritas - Sistema de Ponto Eletrônico com Biometria

O Veritas é um sistema completo de controle de acesso e gestão de presença, utilizando tecnologia biométrica para garantir segurança e precisão. Ele é composto por um dispositivo de hardware inteligente (baseado em ESP8266 com display LCD) e uma aplicação desktop intuitiva para gerenciamento.

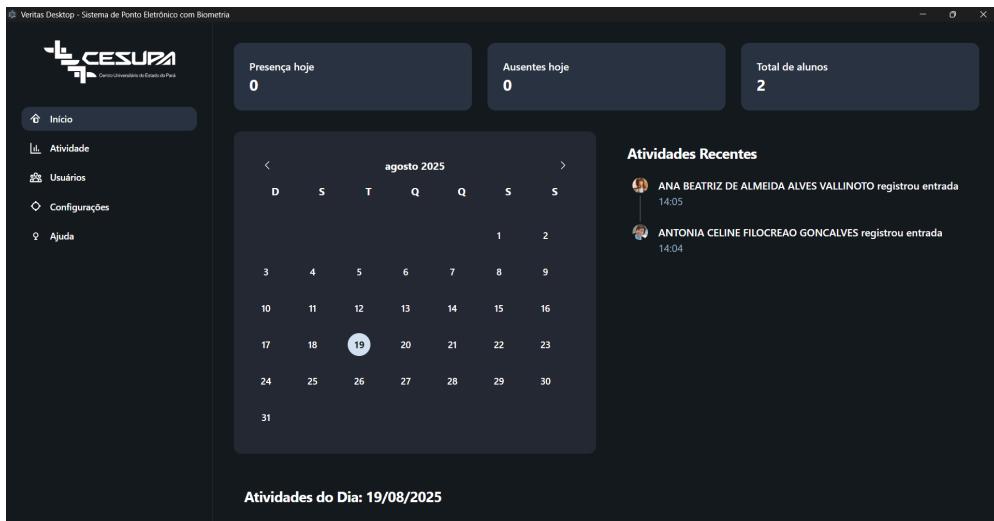


Figura 1: Dashboard principal da aplicação Veritas.

Visão Geral e Funcionamento do Dashboard

A tela principal da aplicação é um dashboard interativo projetado para fornecer uma visão completa e em tempo real do sistema:

- **Cards de Resumo:** No topo, três cards exibem as estatísticas mais importantes do dia: o número de **presentes**, **ausentes** e o **total de alunos** cadastrados. Estes números são atualizados dinamicamente.
- **Calendário Interativo:** Permite selecionar qualquer dia para visualizar o histórico de atividades, facilitando a consulta de registros passados.
- **Feed de Atividades Recentes:** Uma lista mostra as últimas interações com o sensor (entradas e saídas) assim que acontecem.
- **Detalhes do Dia:** Ao clicar em uma data no calendário, a seção de atividades é atualizada para mostrar todos os registros daquele dia específico.

Com o Veritas, você pode facilmente:

- Cadastrar novos usuários com suas impressões digitais.
- Monitorar as atividades de entrada e saída em tempo real.
- Gerenciar os dados dos usuários (editar, excluir, sincronizar), incluindo o sexo para mensagens personalizadas.
- Exportar relatórios de atividade para análise.
- Acessar uma interface web para monitoramento remoto.

Arquitetura do Sistema

O projeto é dividido em duas partes principais: o **Firmware** para o dispositivo embarcado e a **Aplicação Desktop** para o controle e gerenciamento.

1. Hardware

O dispositivo de captura biométrica é montado com os seguintes componentes:

- **Microcontrolador:** NodeMCU (ESP8266) - Um microcontrolador Wi-Fi de baixo custo que combina o chip ESP8266 com uma interface USB. Possui GPIO (pinos de entrada/saída de propósito geral), suporte a Wi-Fi 802.11 b/g/n, e opera a 80MHz (pode chegar a 160MHz). Possui 4MB de memória flash e uma arquitetura de 32 bits que facilita a comunicação sem fio e o desenvolvimento de aplicações IoT.

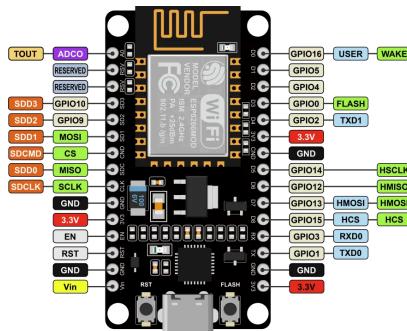


Figura 2: Módulo NodeMCU ESP8266 e suas portas.

- **Sensor Biométrico:** R307 - Um sensor biométrico utilizado para captura e reconhecimento de impressões digitais. Este módulo óptico oferece uma solução robusta para sistemas de autenticação baseados em biometria.

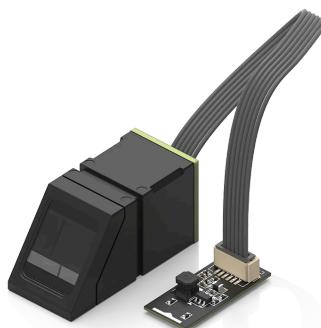


Figura 3: Sensor biométrico R307

Características do Sensor R307:

- **Tecnologia óptica:** Captura imagens de alta resolução das impressões digitais.
- **Comunicação serial:** Utiliza protocolo TTL para comunicação com o microcontrolador.
- **Capacidade de armazenamento:** Permite armazenar até 1000 templates de impressões digitais.
- **Baixo consumo de energia:** Ideal para aplicações embarcadas com restrições de energia.
- **Taxa de reconhecimento:** Menos de 0,5 segundos para identificação.
- **Taxa de falsos positivos:** Menos de 0,001% (FAR).
- **Taxa de falsos negativos:** Menos de 0,1% (FRR).

Funcionamento:

O sensor R307 funciona através de um processo de quatro etapas principais:

1. **Captura da imagem:** Quando um dedo é posicionado sobre o sensor óptico, ele captura uma imagem digital da impressão digital.

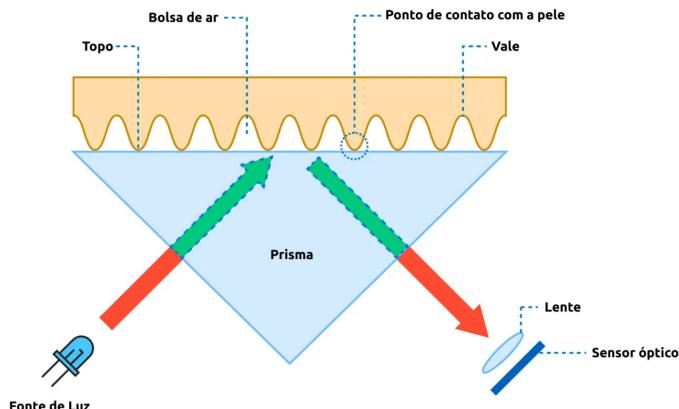


Figura 4: Princípio de Funcionamento do Leitor Óptico de Impressão Digital

2. **Processamento da imagem:** O processador interno do sensor converte a imagem em um template digital único (conjunto de características da impressão).
3. **Armazenamento/Comparação:** Durante o cadastro, o template é armazenado com um ID associado. Durante a verificação, o template capturado é comparado com os já armazenados.
4. **Resultado:** O sensor retorna o ID correspondente em caso de correspondência positiva ou um código de erro caso contrário.

No sistema Veritas, o sensor R307 é integrado através de comunicação serial com o ESP8266, permitindo adicionar ou verificar usuários conforme necessário. Quando uma impressão digital é reconhecida, o sistema consulta o banco de dados para identificar o usuário e registrar sua presença.

- **Módulo de Relógio (RTC): DS3231**

O módulo DS3231 é um relógio de tempo real (RTC) de alta precisão que utiliza comunicação I2C. É crucial para o sistema Veritas, pois:

- **Precisão de tempo:** Possui um cristal oscilador de temperatura compensada que garante medições de tempo extremamente precisas.
- **Bateria de backup:** Contém um slot para bateria CR2032 que mantém o relógio funcionando mesmo quando o dispositivo principal está desligado.
- **Baixo consumo:** Consome pouca energia, ideal para aplicações com restrições energéticas.
- **Funcionalidades:** Fornece dados de segundo, minuto, hora, dia da semana, dia do mês, mês e ano com compensação para anos bissextos.



Figura 5: Módulo RTC 3231

No sistema Veritas, o RTC DS3231 é essencial para registrar com precisão os horários de entrada e saída, garantindo a integridade dos dados de ponto mesmo em casos de queda de energia ou reinicialização do sistema.

- **Display:** LCD 20×4 com módulo I2C



Figura 5: Display LCD 20×4 com módulo I2C

O display LCD 20×4 com módulo I2C é uma tela de cristal líquido que oferece 4 linhas de 20 caracteres cada, totalizando 80 caracteres de espaço para exibição de informações. No sistema Veritas, este componente é fundamental para:

- **Interface com o usuário:** Exibe instruções, mensagens de boas-vindas personalizadas e feedback sobre as operações realizadas.
- **Visualização de status:** Mostra a data e hora atual, o status de conexão e o modo de operação do dispositivo.
- **Confirmação de registro:** Após uma leitura bem-sucedida de impressão digital, exibe o nome e outras informações do usuário identificado.
- **Mensagens de erro:** Comunica problemas de leitura ou falhas no sistema de forma clara e visível.

O módulo I2C acoplado ao display reduz significativamente o número de pinos necessários para a conexão com o microcontrolador - utilizando apenas os pinos SDA e SCL, além da alimentação. Esta economia de pinos é especialmente importante em projetos com o ESP8266, que possui um número limitado de GPIOs disponíveis.

Montagem do Circuito

Os componentes devem ser conectados ao NodeMCU/ESP8266 da seguinte forma:

Componente	Pino no ESP8266
Sensor Biométrico R307	
VCC (5V)	VU (5V)
GND	GND
TX (Sensor)	D7 (GPIO 13)
RX (Sensor)	D6 (GPIO 12)
TOUCH (Sensor)	D5 (GPIO 14)
Módulo RTC DS3231	
VCC (3.3V)	3V3
GND	GND
SDA	D3 (GPIO 0)
SCL	D4 (GPIO 2)
Display LCD I2C 20×4	
VCC (5V)	VU (5V)
GND	GND
SDA	D3 (GPIO 0)
SCL	D4 (GPIO 2)

Nota: Os pinos SDA e SCL do RTC e do Display LCD são conectados nos mesmos pinos I2C do ESP8266.

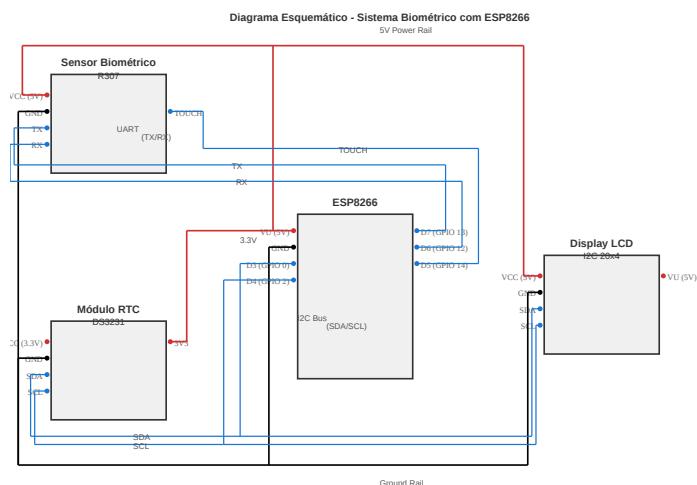


Figura 6: Diagrama esquemático do Veritas.



Figura 7: Veritas instalado no NPJ.

2. Software

a. Firmware (Dispositivo Embarcado)

O cérebro do dispositivo de hardware é programado em C++ utilizando o framework Arduino através do PlatformIO.

- **Linguagem:** C++ (Arduino)
- **Plataforma:** PlatformIO
- **Bibliotecas Principais:**
 - [Adafruit_Fingerprint](#) : Para comunicação com o sensor biométrico.
 - [RTClib](#) : Para obter a data e hora do módulo RTC.
 - [LiquidCrystal_I2C](#) : Para controlar o display LCD.
 - [SoftwareSerial](#) : Para criar uma porta serial secundária para o sensor.
 - [ArduinoJson](#) : Para a comunicação serial com a aplicação desktop via JSON.
- **Funcionalidades:**
 - Gerencia o cadastro, verificação e exclusão de impressões digitais.
 - Exibe informações no display LCD (tela padrão, relógio, mensagens de status e erro).
 - Solicita dados do usuário (nome, sexo) à aplicação desktop após uma leitura bem-sucedida para exibir uma mensagem de boas-vindas personalizada.
 - Envia os registros de atividade com timestamp preciso do RTC para a aplicação desktop.

b. Aplicação Desktop (Painel de Controle)

A interface de gerenciamento é uma aplicação desktop multiplataforma construída com Electron.

- **Tecnologias:** Electron, Node.js, HTML5, Tailwind CSS, JavaScript.
- **Localização dos Arquivos:** O código-fonte da aplicação desktop está na pasta [/data](#).
- **Funcionalidades:**
 - Interface gráfica para listar, cadastrar, editar e excluir usuários.
 - Responde às solicitações de dados do dispositivo ESP para a exibição no display.
 - Visualização da tabela de atividades em tempo real.
 - Dashboard com estatísticas de presença.
 - Configuração da porta de comunicação com o hardware.

- Exportação de dados em formato CSV.
- Servidor web local para acesso remoto ao dashboard.
- **Comunicação:** Utiliza a biblioteca `serialport` para se comunicar com o dispositivo ESP.
- **Banco de Dados:** Um arquivo `database.json` local (na pasta de dados do usuário do sistema operacional) armazena os dados de usuários e atividades.

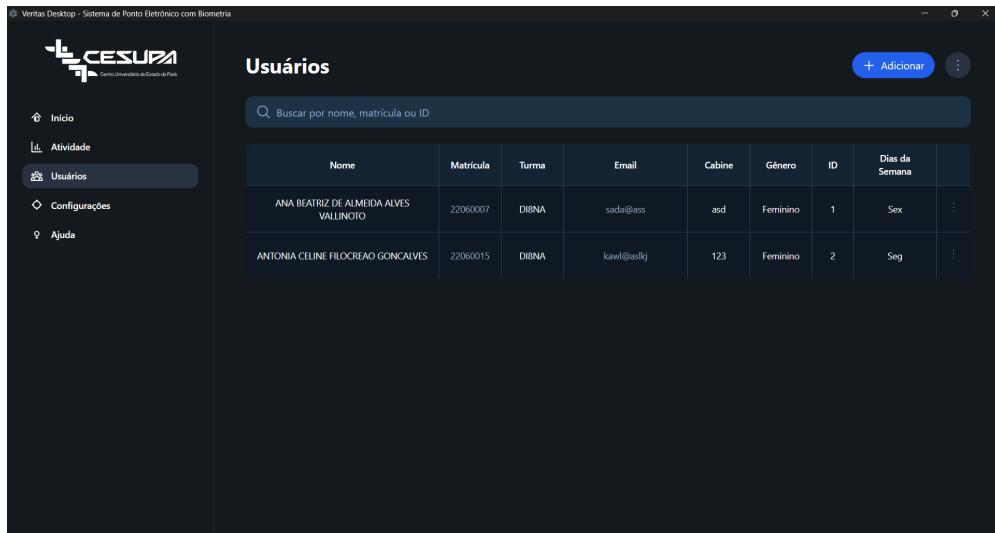


Figura 4: Tela de gerenciamento de usuários.

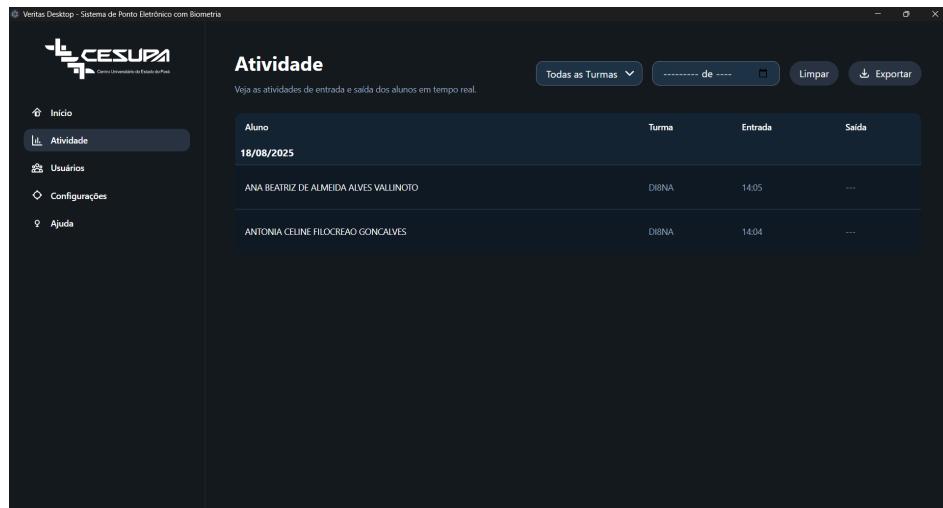


Figura 5: Tela de visualização de atividades.

Instalação e Uso

Siga os passos abaixo para configurar e executar o sistema.

1. Pré-requisitos

- Visual Studio Code com a extensão PlatformIO IDE.
- Node.js (versão 14 ou superior).
- Hardware montado conforme a especificação acima.

2. Configurar o Firmware

1. Abra a pasta do projeto no Visual Studio Code.

2. Conecte o dispositivo ESP8266 ao computador via USB.
3. O PlatformIO deverá reconhecer o projeto automaticamente.
4. Para compilar e enviar o firmware para o dispositivo, execute o seguinte comando no terminal do PlatformIO:

```
platformio run --target upload
```

3. Configurar a Aplicação Desktop

1. Navegue até a pasta da aplicação:

```
cd data
```

2. Instale as dependências do Node.js:

```
npm install
```

3. Inicie a aplicação:

```
npm start
```

4. Primeiro Uso

1. Ao abrir a aplicação, vá para a aba **Configurações**.
2. Selecione a porta serial correspondente ao seu dispositivo ESP na lista e clique em **Salvar configurações**. O indicador de status deve ficar verde.
3. Vá para a aba **Usuários** e clique no botão **Adicionar Usuário** para começar a cadastrar as pessoas. Preencha todos os campos e siga as instruções na tela para o registro da impressão digital.

Como Construir (Build)

Para empacotar a aplicação desktop em um instalador para Windows, macOS ou Linux, utilize o seguinte comando na pasta `/data` :

```
npm run dist
```

Os arquivos de saída serão gerados na pasta `/build`.

O código fonte do Veritas está disponível para consulta em <https://github.com/geohonorato/Veritas>

Autor

- **Geovanni Honorato**

Licença

Este projeto está licenciado sob a licença ISC.