SISTEMA DE GERENCIAMENTO DE PONTO ELETRÔNICO E BANCO DE HORAS

Enrico Cabana Nascimento, Igor Garcez de Oliveira, João Paulo dos Santos Felix, Alberson Wander Sá dos Santos, Hélio Lourenço Esperidião Ferreira.

Colégios Univap Unidade Centro Fundação Valeparaibana de Ensino, Rua Paraibuna, 75, 12245-020, São José dos Campos-SP, Brasil, enricocabana2009@gmail.com

Resumo

O banco de horas é um mecanismo de compensação de jornada previsto na CLT desde 1998, mas os métodos manuais de controle de ponto são vulneráveis a fraudes e imprecisões. Este trabalho teve como objetivo propor e desenvolver, para a empresa IF Informática, um sistema de gerenciamento de ponto eletrônico do tipo e-Time, voltado ao registro e controle das horas trabalhadas em ambiente organizacional, com automação dos cálculos de jornada, controle de horas extras e gerenciamento integrado do banco de horas. O sistema foi desenvolvido utilizando JavaScript (Node.js e Express) na camada de aplicação e MySQL como banco de dados relacional, adotando arquitetura em camadas (apresentação, aplicação e dados). Com base no desenvolvimento realizado, o objetivo proposto foi alcançado: a solução contempla o registro das marcações de ponto, o cálculo automático da jornada, o controle de horas extras e do banco de horas, a geração de relatórios gerenciais e mecanismos de auditoria, reduzindo falhas decorrentes de processos manuais e contribuindo para uma gestão mais segura das informações trabalhistas.

Palavras-chave: Banco de horas. Ponto eletrônico. Jornada de trabalho.
Área do Conhecimento: Ciências Exatas e da Terra / Ciência da Computação.
Curso: Técnico em Informática

Introdução

O banco de horas trabalhista surgiu em 1998, por meio da Lei nº 9.601, que alterou o artigo 59 da Consolidação das Leis do Trabalho, em época de grave crise econômica que gerou a demissão de muitos trabalhadores e o fechamento de empresas, principalmente no ramo das montadoras e das metalúrgicas. O instituto funciona como uma conta corrente de horas: o empregado que trabalha em hora extraordinária tem saldo positivo, para compensar depois, e aquele que trabalha menos tem saldo negativo, devendo trabalhar mais para pagar as horas (Pontotel, 2026).

Atualmente, o método de controle é muito vulnerável e suscetível a falhas, pois há margem para alterações ou preenchimento impreciso, o que pode gerar saldo indevido no banco de horas (Ferreira, 2017). Esse cenário torna-se ainda mais relevante considerando que a legislação trabalhista brasileira determina que empresas com mais de 10 empregados devem obrigatoriamente controlar a jornada de trabalho de seus funcionários; a ausência de sistemas mais robustos pode dificultar a fiscalização e abrir margem para irregularidades que impactam tanto os direitos dos trabalhadores quanto a gestão interna das organizações (Lira; Schiavon; Freguglia, 2022).

Com a implantação de sistemas eletrônicos de controle de horas trabalhadas, torna-se possível gerenciar com precisão a rotina dos funcionários, armazenando os registros e realizando o cálculo de horas trabalhadas, horas excedentes, horas noturnas e horas faltantes, além da emissão de relatórios para consultas futuras e compensação de horas (Ferreira, 2017).

Nesse contexto, o presente trabalho tem como objetivo propor e desenvolver, em parceria com a empresa IF Informática, um sistema de gerenciamento de ponto eletrônico do tipo e-Time, voltado ao registro e controle das horas trabalhadas de seus colaboradores. Atualmente, o registro é feito em um relógio de ponto mecânico de cartão, cujos dados impressos são transcritos manualmente pelo setor de Recursos Humanos para planilhas eletrônicas, processo sujeito a erros de digitação e retrabalho. Diante disso, a proposta consiste na criação de uma solução tecnológica capaz de registrar as marcações de ponto dos colaboradores, realizar o cálculo automático da jornada de trabalho, controlar horas extras e gerenciar o banco de horas de forma integrada, utilizando JavaScript como linguagem principal da API.

Metodologia

O sistema foi desenvolvido utilizando JavaScript como linguagem principal, por meio do ambiente de execução Node.js e do micro-framework Express.js, responsável pelo roteamento de dados e requisições HTTP (Mardan, 2014). O armazenamento dos dados foi realizado em um Sistema Gerenciador de Banco de Dados Relacional (SGBDR) MySQL (Elmasri; Navathe, 2011). A interface do usuário foi construída em HTML5 e CSS3 (Duckett, 2014).

A arquitetura do sistema foi definida em três camadas principais: apresentação, aplicação e dados. A camada de apresentação é responsável pela interação com o usuário, permitindo o registro de ponto, a consulta de informações e a visualização de dados processados. A camada de aplicação concentra as regras de negócio, sendo responsável pelo processamento das informações e pela execução automatizada de cálculos, validações e atualizações. A camada de dados é responsável apenas pela persistência e integridade das informações.

Internamente, a camada de aplicação foi organizada em seis componentes principais: *Router, Middleware, Controller, Service, Model* e *DAO* (*Data Access Object*). As rotas (*Router*) recebem as requisições HTTP e as encaminham, passando primeiro pelos Middlewares, responsáveis por validar o token de autenticação (JWT) e as permissões do usuário (perfil Administrador ou Funcionário) antes de liberar o acesso ao recurso solicitado. Em seguida, os Controllers interpretam a requisição, validam os dados de entrada e delegam o processamento ao Service correspondente, funcionando como uma camada fina de orquestração. A lógica de negócio propriamente dita, como o cálculo da jornada, a apuração do saldo do banco de horas e as demais validações, é concentrada nos Services, responsáveis por aplicar as regras definidas para cada operação do sistema. Os dados manipulados pelo sistema são representados pelos Models, que definem a estrutura das entidades do domínio (usuário, marcação de ponto, ocorrência etc.) utilizadas pelas demais camadas. O acesso ao banco de dados MySQL não é feito diretamente pelos Services, mas por meio de objetos DAO, responsáveis exclusivamente por executar as consultas SQL e retornar os dados já estruturados para a camada de aplicação. Essa separação entre Middleware, Controller, Service, Model e DAO reduz o acoplamento entre a autenticação, a interpretação da requisição, a lógica de negócio e o acesso a dados, facilitando a manutenção, os testes e eventuais alterações no esquema do banco sem impactar as regras já implementadas. Nesse arranjo, a interface web, construída em HTML5 e CSS3, consome os dados retornados pela API em formato JSON.

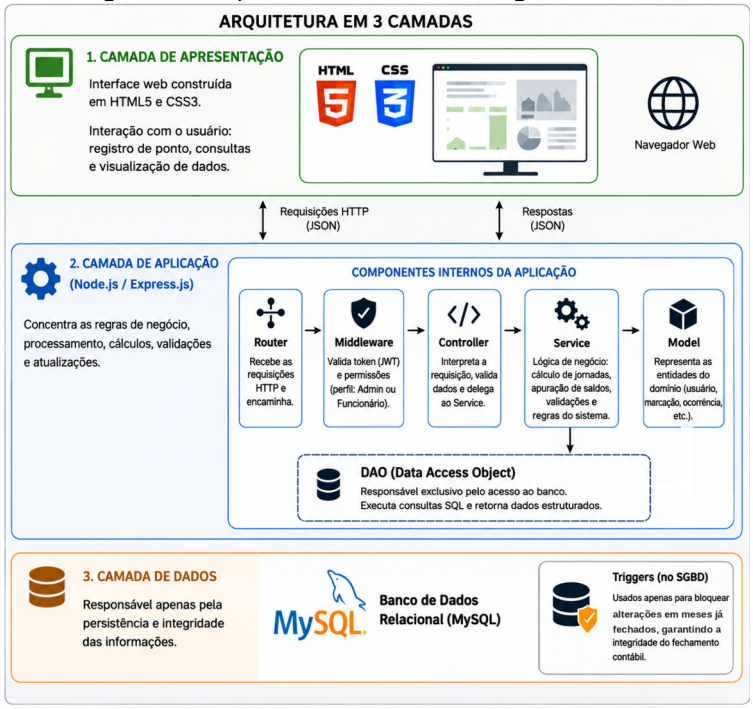
O banco de dados do sistema está organizado em módulos funcionais: estrutura organizacional (setores, tipos_ocorrencia, feriados), usuários e dados pessoais (usuarios, documentos, enderecos), movimentação (tabela ponto, que registra as quatro marcações diárias e os campos meta_do_dia, total_dia_minutos e saldo_dia_minutos, e tabela solicitacoes, que registra os pedidos de ajuste de ponto feitos pelos funcionários, com um arquivo anexo opcional como comprovante, e sua aprovação ou recusa pelo RH), gestão e auditoria (fechamentos_mensais, auditoria) e comunicação (avisos). O cálculo das horas trabalhadas é realizado na camada de aplicação (Node.js), enquanto o banco de dados utiliza triggers exclusivamente para bloquear alterações em meses já fechados, garantindo a integridade do fechamento contábil sem acoplar regra de negócio ao SGBD. O sistema conta ainda com um módulo de backup, que gera um arquivo compactado com os dados exportados e um resumo criptográfico (hash SHA-256) para verificação posterior de integridade.

A autenticação do sistema é realizada por meio de tokens JWT (*JSON Web Token*), gerados no momento do login e enviados pelo cliente em requisições subsequentes por meio do cabeçalho *Authorization*. Um middleware dedicado valida o token em cada requisição protegida, verificando sua assinatura e validade e recusando o acesso (código HTTP 401) em caso de token ausente, expirado ou inválido. As *claims* do token, que incluem e-mail, perfil (Administrador ou Funcionário) e identificador do colaborador, permitem que o sistema aplique controle de acesso baseado em perfil. As senhas dos usuários são armazenadas com hash gerado pelo algoritmo bcrypt, nunca em texto puro.

A validação do sistema foi realizada por meio de testes manuais, nos quais diferentes cenários de marcação de ponto (jornadas completas, horas extras, faltas parciais e ocorrência de feriados) foram inseridos no sistema e seus resultados de cálculo total de horas trabalhadas, saldo diário e saldo acumulado no banco de horas foram comparados com os valores obtidos por meio de cálculo manual, realizado em paralelo com base nas mesmas regras de jornada. A convergência entre os valores calculados pelo sistema e os valores de referência calculados manualmente foi utilizada como critério para validar a correção da lógica implementada na camada de aplicação. Para a validação

funcional do sistema, foram utilizados exclusivamente dados simulados de colaboradores fictícios e jornadas hipotéticas, sem a coleta, armazenamento ou processamento de dados pessoais de funcionários reais da organização parceira, o que isenta esta pesquisa de submissão ao Comitê de Ética em Pesquisa (CEP). A Figura 1 apresenta uma síntese da arquitetura do sistema, organizada em três camadas: apresentação, aplicação e dados. A camada de apresentação é responsável pela interação com o usuário por meio da interface web. A camada de aplicação concentra o processamento das requisições e as regras de negócio, seguindo o fluxo entre Router, Middleware, Controller, Service e Model, sendo o Service responsável pelo acesso ao DAO. Por fim, a camada de dados utiliza o MySQL para armazenar e garantir a integridade das informações.

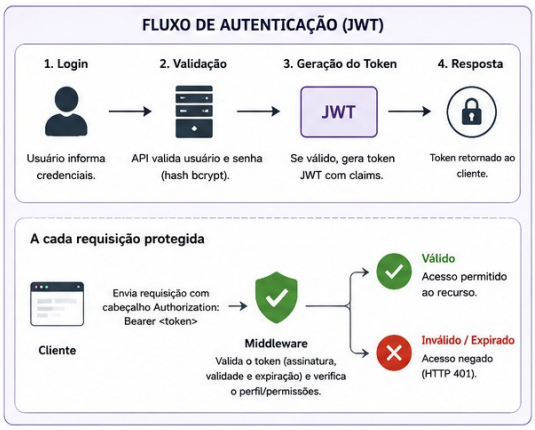
Figura 1 - Arquitetura do sistema de gestão de horas.



Fonte: Autores, 2026.

A autenticação de usuários é um fator muito importante para a segurança das aplicações de forma geral. A Figura 2 apresenta o fluxo de autenticação utilizando Token JWT, desde a validação das credenciais e geração do token até sua utilização em requisições protegidas, permitindo ou negando o acesso aos recursos da aplicação.

Figura 2 - Arquitetura do sistema de autenticação da aplicação.

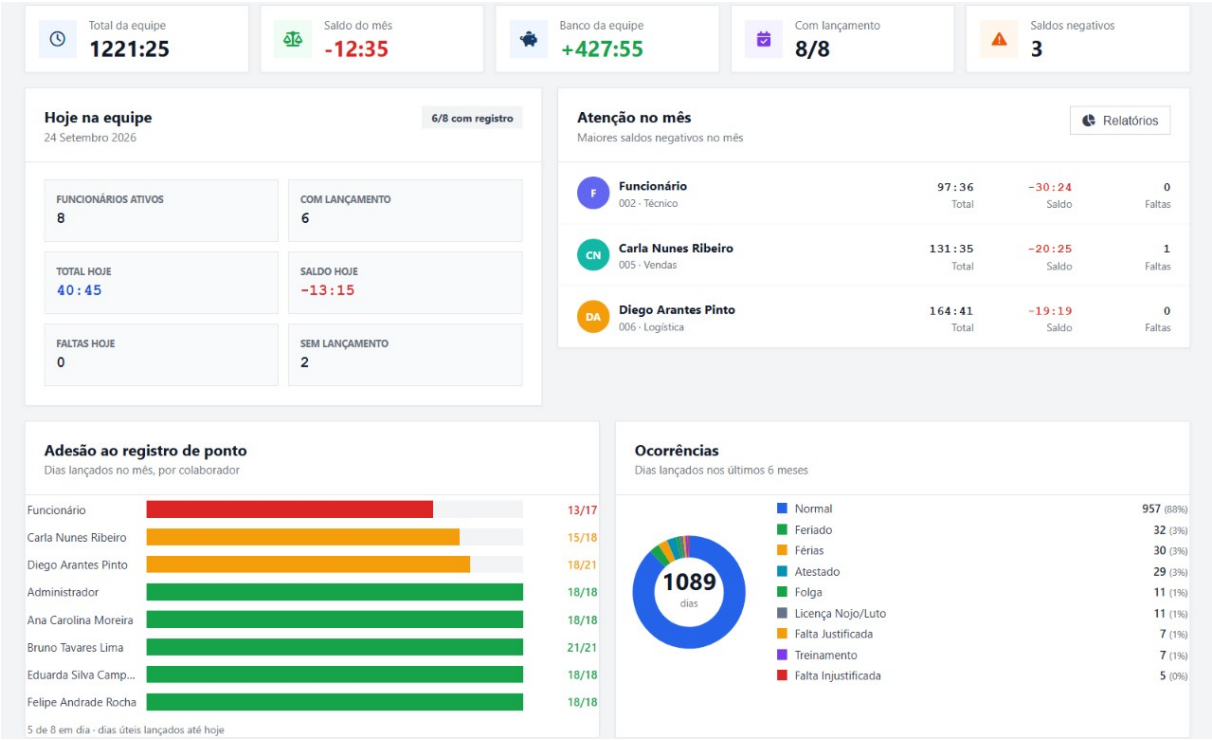


Fonte: Autores, 2026.

Resultados

A interface gráfica de Dashboard (Visão Geral) do sistema, voltada ao gestor da equipe, apresenta no topo indicadores consolidados do mês corrente: o total de horas trabalhadas pela equipe (1221:25), o saldo do mês (-12:35), o saldo acumulado no banco de horas da equipe (+427:55), a proporção de funcionários com lançamento de ponto realizado (8/8) e a quantidade de funcionários com saldo negativo no período (3). Logo abaixo, o painel "Hoje na equipe" resume a situação do dia corrente, informando o número de funcionários ativos (8), quantos já registraram o ponto (6), o total de horas (40:45) e o saldo (-13:15) apurados no dia, além das faltas (0) e dos lançamentos pendentes (2). Ao lado, o painel "Atenção no mês" lista os funcionários com os maiores saldos negativos, exibindo o total de horas trabalhadas, o saldo e o número de faltas de cada um. Na parte inferior, dois gráficos complementam a análise: a adesão ao registro de ponto, que compara os dias úteis já decorridos com os efetivamente lançados por colaborador, e a distribuição das ocorrências registradas nos últimos meses, conforme a Figura 3.

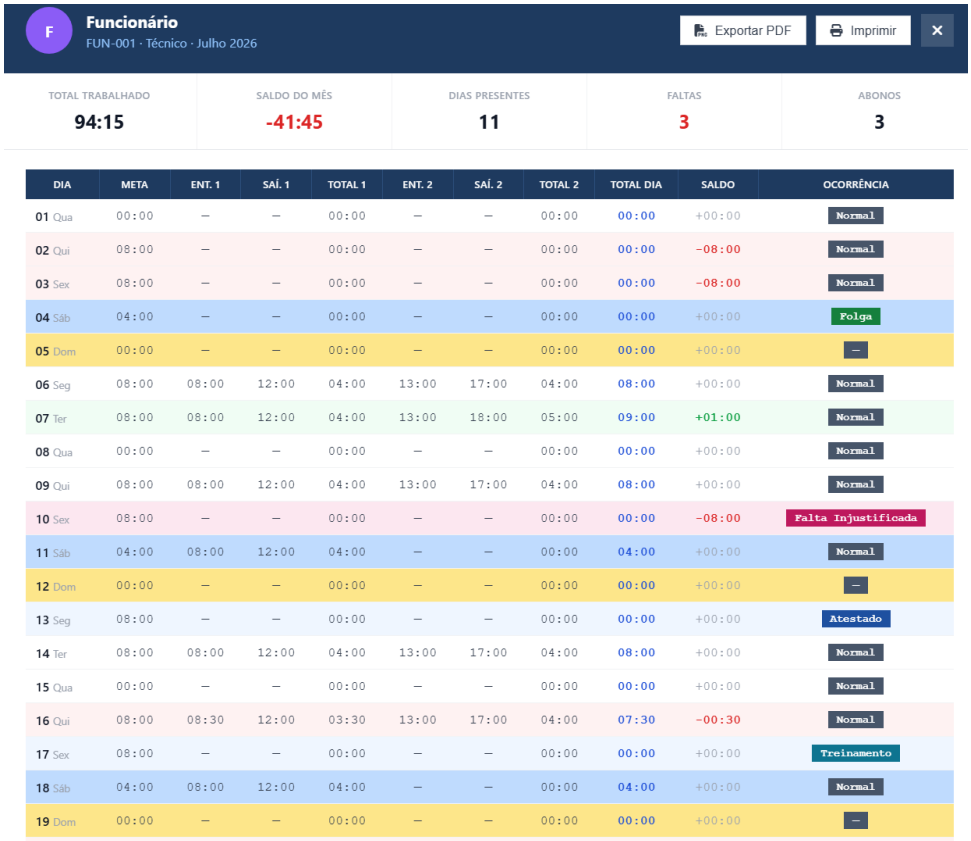
Figura 3 - Dashboard (Visão Geral) do sistema.



Fonte: Autores, 2026.

A interface gráfica de espelho de ponto individual (relatório mensal) apresenta o detalhamento da jornada de um colaborador ao longo do mês selecionado, no caso, um funcionário do setor Técnico, referente a julho de 2026. No cabeçalho são exibidos os indicadores consolidados do período: total de horas trabalhadas (94:15), saldo do mês (-41:45), dias presentes (11), faltas (3) e abonos (3). Abaixo, uma tabela detalha o registro dia a dia, com a meta de horas, os quatro horários de marcação (entrada e saída dos períodos matutino e vespertino), o total de horas apuradas no dia, o saldo diário resultante e a ocorrência associada, como "Normal", "Falta Injustificada", "Falta Justificada", "Atestado", "Folga", "Treinamento" ou "Licença Nojo/Luto", permitindo à gestão visualizar de forma clara os motivos por trás de cada saldo positivo, negativo ou neutro registrado, conforme a Figura 4.

Figura 4 - Relatório mensal individual (espelho de ponto).



Funcionário
FUN-001 · Técnico · Julho 2026

Exportar PDF | Imprimir

TOTAL TRABALHADO	SALDO DO MÊS	DIAS PRESENTES	FALTAS	ABONOS
94:15	-41:45	11	3	3

DIA	META	ENT. 1	SAÍ. 1	TOTAL 1	ENT. 2	SAÍ. 2	TOTAL 2	TOTAL DIA	SALDO	OCORRÊNCIA
01 Qua	00:00	–	–	00:00	–	–	00:00	00:00	+00:00	Normal
02 Qui	08:00	–	–	00:00	–	–	00:00	00:00	-08:00	Normal
03 Sex	08:00	–	–	00:00	–	–	00:00	00:00	-08:00	Normal
04 Sáb	04:00	–	–	00:00	–	–	00:00	00:00	+00:00	Folga
05 Dom	00:00	–	–	00:00	–	–	00:00	00:00	+00:00	–
06 Seg	08:00	08:00	12:00	04:00	13:00	17:00	04:00	08:00	+00:00	Normal
07 Ter	08:00	08:00	12:00	04:00	13:00	18:00	05:00	09:00	+01:00	Normal
08 Qua	00:00	–	–	00:00	–	–	00:00	00:00	+00:00	Normal
09 Qui	08:00	08:00	12:00	04:00	13:00	17:00	04:00	08:00	+00:00	Normal
10 Sex	08:00	–	–	00:00	–	–	00:00	00:00	-08:00	Falta Injustificada
11 Sáb	04:00	08:00	12:00	04:00	–	–	00:00	04:00	+00:00	Normal
12 Dom	00:00	–	–	00:00	–	–	00:00	00:00	+00:00	–
13 Seg	08:00	–	–	00:00	–	–	00:00	00:00	+00:00	Atestado
14 Ter	08:00	08:00	12:00	04:00	13:00	17:00	04:00	08:00	+00:00	Normal
15 Qua	00:00	–	–	00:00	–	–	00:00	00:00	+00:00	Normal
16 Qui	08:00	08:30	12:00	03:30	13:00	17:00	04:00	07:30	-00:30	Normal
17 Sex	08:00	–	–	00:00	–	–	00:00	00:00	+00:00	Treinamento
18 Sáb	04:00	08:00	12:00	04:00	–	–	00:00	04:00	+00:00	Normal
19 Dom	00:00	–	–	00:00	–	–	00:00	00:00	+00:00	–

Fonte: Autores, 2026.

Discussão

A escolha de concentrar o cálculo das horas trabalhadas na camada de aplicação (Node.js), e não no banco de dados, foi a decisão de arquitetura mais relevante do projeto: o banco de dados manteve um papel de persistência e proteção de integridade, por meio de triggers que bloqueiam alterações em meses já fechados, enquanto a API permaneceu responsável pelas regras de negócio e pelos cálculos, como o cômputo da jornada diária, a apuração de horas extras e a atualização do saldo do banco de horas. Essa separação contribui para a redução de erros e para a centralização da lógica do sistema, evitando inconsistências no tratamento dos dados. Por outro lado, essa concentração de regras na camada de aplicação também traz possíveis problemas: como o SGBD não valida as regras de cálculo por conta própria, uma falha de implementação ou uma alteração indevida no código da API poderia gerar saldos incorretos no banco de horas sem que o banco de dados identificasse a inconsistência, já que os triggers atuam apenas sobre o bloqueio de meses fechados, e não sobre a correção aritmética dos valores. Esse ponto reforça a importância de testes e revisões cuidadosos sobre essa camada, especialmente antes de cada fechamento mensal.

Quanto à redução de falhas em processos manuais e à segurança da informação, esse objetivo foi reforçado pela separação de perfis de acesso (RH e Funcionário), pelo módulo de Auditoria que registra toda alteração realizada no sistema, com valor antigo e novo, e pelo bloqueio de edições em períodos já fechados. Essas características são particularmente relevantes em um sistema que lida com direitos trabalhistas, já que dificultam adulterações indevidas nos registros de ponto e aumentam a rastreabilidade das informações em relação a um controle manual ou a uma jornada fixa e genérica.

Comparando a proposta com soluções comerciais já consolidadas no mercado, observam-se perfis distintos de sistemas de controle de ponto. O PontoSimples (Pontosimples, 2026) opera em modelo SaaS corporativo, com marcação via aplicativo móvel ou web, conformidade com a Portaria nº 671/2021 do Ministério do Trabalho, gestão de banco de horas com alertas de saldo, aprovação de solicitações

de abono e ajuste de ponto pelo gestor e exportação do fechamento em formatos compatíveis com sistemas de folha de pagamento, funcionalidades convergentes com as do sistema aqui desenvolvido, especialmente o módulo de solicitações de ajuste implementado no CronaSys, no qual o funcionário registra o pedido e o RH aprova ou nega. Já o Meu Controle de Ponto (Meu Controle de Ponto, 2026) representa um perfil mais simples e de uso individual, em que o próprio usuário registra sua jornada, folgas e horas extras e gera relatórios pessoais, sem módulos de gestão corporativa, aprovação por terceiros ou auditoria, o que evidencia que as soluções de mercado variam desde ferramentas pessoais e enxutas até plataformas corporativas completas. Por outro lado, o sistema proposto no presente trabalho concentra-se em um subconjunto mais específico dessas funcionalidades: o registro das quatro marcações diárias, o cálculo de jornada e do banco de horas, o módulo de solicitações de ajuste e o módulo de auditoria com bloqueio de meses fechados, sem abranger integrações externas como folha de pagamento ou eSocial, nem dispositivos de captura biométrica ou geolocalização, o que caracteriza uma solução mais enxuta e voltada ao aprendizado da arquitetura de um sistema desse tipo. Ainda assim, o mecanismo de auditoria com registro de valor antigo e novo a cada alteração, presente no sistema desenvolvido, representa um diferencial de rastreabilidade em relação a soluções que priorizam a automação operacional em detrimento do detalhamento de cada alteração realizada.

Conclusão

Diante do exposto, conclui-se que o desenvolvimento do sistema atendeu ao objetivo proposto, oferecendo à IF Informática uma alternativa tecnológica capaz de substituir os métodos manuais de controle de ponto por um processo automatizado, mais preciso e seguro. A centralização das regras de cálculo na camada de aplicação, aliada aos mecanismos de auditoria e de bloqueio de edições em períodos já fechados, demonstra que é possível conciliar flexibilidade no registro da jornada com a integridade exigida pelas obrigações trabalhistas. Como sugestão para trabalhos futuros, destaca-se a possibilidade de integrar métodos de autenticação biométrica ou reconhecimento por imagem às marcações de ponto, ampliando ainda mais a confiabilidade do sistema.

Referências

DUCKETT, Jon. HTML e CSS: projete e construa websites. Rio de Janeiro: Alta Books, 2014.

ELMASRI, Ramez; NAVATHE, Shamkant B. Sistemas de banco de dados. 6. ed. São Paulo: Pearson Addison Wesley, 2011.

FERREIRA, Marcos Thiago. O uso da biometria no controle das horas trabalhadas pelos policiais civis de Santa Catarina e seus reflexos jurídicos. 2017. Monografia (Especialização) – Universidade Federal de Santa Catarina, Araranguá, 2017.

LIRA, Ísis Ferreira; SCHIAVON, Laura de Carvalho; FREGUGLIA, Ricardo da Silva. Regulação eletrônica da jornada de trabalho e comportamento das empresas: evidências para o Brasil. In: ENCONTRO NACIONAL DE ECONOMIA, 2022.

MARDAN, Azat. Pro Express.js. Apress, 2014.

MEU CONTROLE DE PONTO. Controle pessoal de horas trabalhadas. Disponível em: https://www.meucontroledeponto.com.br/. Acesso em: 22 ago. 2026.

PONTOSIMPLES. Controle de ponto eletrônico online. Disponível em: https://pontosimples.com/. Acesso em: 22 ago. 2026.

PONTOTEL. Banco de horas: o que é, como funciona, regras da CLT e como implementar na empresa. Disponível em: https://www.pontotel.com.br/banco-de-horas/. Acesso em: 22 ago. 2026.