**Documento de Requisitos de Software**

**E-Ticket**

1.1

**Equipe**

Daniel Elias

João Félix

Joceli Franco

Mateus Almeida

**Rio Branco – AC**

**2024**

**Histórico de Alterações**

| **Data** | **Versão** | **Descrição** | **Autor** |
| --- | --- | --- | --- |
| 21/11/2023 | 1.0 | Criação da documentação de requisitos. | Mateus |
| 20/02/2024 | 1.1 | Adição e especificando de novos requisitos funcionais, atualização da descrição do projeto e inclusão de cronograma de sprints. | João Félix e Daniel Elias |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. **Análise do Problema**

O projeto em questão visa desenvolver um software para otimizar o processo de compra de tickets para as refeições do Restaurante Universitário (RU) de uma universidade.

Atualmente, o sistema de compra de tickets é manual, exigindo que os usuários enfrentem filas para adquirir os tickets impressos. Após a compra, outra fila é enfrentada na entrada do restaurante, onde os tickets são verificados manualmente, juntamente com a identificação do usuário.

O processo de compra e autenticação não é integrado, e o pagamento é feito exclusivamente em dinheiro ou via Pix. A contagem dos tickets é realizada manualmente na entrada do restaurante. O objetivo do projeto é virtualizar esse processo, permitindo que os usuários comprem tickets online dentro de seus limites mensais e utilizem um código digital na entrada do restaurante para autenticação.

1. **Necessidades Básicas do Cliente**

As partes interessadas incluem a Pró-reitoria de Assistência Estudantil (Proaes), a Administração do Restaurante Universitário (RU) e o Núcleo de Tecnologia da Informação (NTI), além dos usuários do restaurante. O software deve atender às necessidades de automatização do processo de compra e autenticação de tickets, fornecer uma interface administrativa para gerenciar benefícios, preços e relatórios, e garantir a segurança e integridade dos dados dos usuários.

1. **Estudo de Viabilidade** 
   1. **Viabilidade Técnica**

A virtualização do processo de compra de tickets e autenticação dos usuários é viável tecnicamente, com a utilização de tecnologias web para desenvolvimento do software, integração com sistemas de pagamento online e implementação de um sistema de autenticação seguro.

* 1. **Viabilidade Econômica**

A implementação do software pode gerar economia de recursos ao reduzir a necessidade de impressão e contagem manual de tickets, além de proporcionar uma experiência mais conveniente aos usuários. No entanto, é necessário avaliar os custos de desenvolvimento e manutenção do sistema em relação aos benefícios esperados.

* 1. **Viabilidade Legal**

É importante considerar questões legais relacionadas à privacidade e proteção de dados dos usuários, bem como a conformidade com regulamentações de pagamento online e políticas de assistência estudantil da universidade.

**4. Missão do Software**

O software tem como missão simplificar e otimizar o processo de compra e autenticação de tickets para refeições no Restaurante Universitário (RU), oferecendo uma solução eficiente e segura para usuários e administradores, promovendo a praticidade e a transparência nas operações do restaurante.

**5. Limites do Sistema**

| **ID** | **Funcionalidade** | **Justificativa** |
| --- | --- | --- |
| **L1** | PagTesouro | Burocracia e legislação ainda a ser estudada para a implementação e entender como a API funciona. |
| **L2** | Bolsas | Como vamos definir se um aluno bolsista vai ter o benefício da gratuidade ou não |

**6. Benefícios Gerais**

| **ID** | **Benefício** |
| --- | --- |
| **B1** | Praticidade e eficiência |
| **B2** | Sustentável |
| **B3** | Não perder os tickets |
| **B4** | Controle de fluxo |
| **B5** | Análise de dados |
| **B6** | Menos desperdício de alimentos |

**7. Restrições**

| **ID** | **Restrição** | **Descrição** |
| --- | --- | --- |
| **R1** | Limitação de espaço | Pode ocorrer a superlotação do local |
| **R2** | Rede | Existe uma dificuldade para a implementação no ponto de acesso de rede na entrada do RU |

**8. Atores**

| **ID** | **Atores** | **Descrição** |
| --- | --- | --- |
| **A1** | Daniel Elias | Back-end e Banco de dados |
| **A2** | João Félix | Back-end e Banco de dados |
| **A3** | Joceli Franco | Front-end e Documentação |
| **A4** | Mateus Almeida | Front-end e APIs |

**9. Requisitos Funcionais**

| **ID** | **Funcionalidade** | **Necessidades** | **Classificação** |
| --- | --- | --- | --- |
| **RF1** | Comprartickets | Usuários precisam de uma maneira eficiente de adquirir tickets de refeição para garantir praticidade e agilidade no acesso às refeições. | Alta |
| **RF2** | Cardápio | Expor o cardápio diário (sujeito a alterações) para os usuários. | Baixa |
| **RF3** | Validação de ticket | Usuários necessitam de um processo eficiente de validação de tickets para garantir o acesso correto e controlado ao Restaurante Universitário. | Alta |
| **RF4** | Método de Pagamento | No início vamos implementar método de pagamento com api de banco para teste, mas depois vamos implementar o PagTesouro | Alta |
| **RF5** | Geração de relatórios e Dashboard | Administradores necessitam de ferramentas eficazes para analisar dados e obter insights sobre o fluxo de pessoas no Restaurante Universitário. | Alta |
| **RF6** | Gestão de usuários | Dividir os usuários dentro do sistema para poder separar funcionalidades de administrador e usuário comum. | Alta |
| **RF7** | Login com IDUfac | Para facilitar o login do usuário vamos implementar o login com o IDUfac | Baixa |
| **RF8** | Regra de negócios IDUfac | Poderíamos solicitar ao NTI que enviasse as informações do IDUfac para que pudéssemos desenvolver regras de negócio destinadas a definir preços conforme o tipo de usuário, a saber: graduação, pós-graduação, doutorado, servidor público e usuário externo.  se é bolsista ou não | Alta |
| **RF9** | Análise preditiva | No restaurante universitário, se a produção de alimentos exceder o consumo das pessoas, haverá desperdício. Portanto, por meio da análise preditiva, será realizada uma avaliação para determinar se o fluxo de pessoas em determinado dia é maior ou não, visando alcançar uma maior eficiência operacional. | Alta |

**10. Requisitos Não-Funcionais**

| **ID** | **Requisitos** | **Categoria** | **Classificação** |
| --- | --- | --- | --- |
| **NRF1** | Não suscetível a queda de rede | Eficiência | Alta |
| **NRF2** | Verificação de tickets em pouco tempo | Desempenho | Alta |
| **NRF3** | Facilidade de Uso | Usabilidade | Média |
| **NRF4** | Proteção de dados | Segurança | Alta |

**11. Requisitos de Hardware**

**11.1. Configuração Mínima**

Roteadores no RU

Nobreak

Computador

**11.2. Configuração Recomendada**

Roteadores no RU funcionando com integridade

Nobreak carregado

Computador funcionando

**12. Ferramentas de Desenvolvimento e Licença de Uso**

As ferramentas selecionadas para o desenvolvimento deste projeto são essenciais para garantir uma implementação eficiente e de alta qualidade. A escolha cuidadosa dessas ferramentas visa proporcionar um ambiente de desenvolvimento colaborativo, ágil e produtivo.

**Git e GitHub:** Utilizaremos o Git como sistema de controle de versão para gerenciar o código-fonte do projeto. O GitHub será nossa plataforma de hospedagem, permitindo o armazenamento centralizado do código, colaboração entre membros da equipe e controle de versão eficaz.

**Django (Python):** Atendendo à solicitação do NTI, optamos pelo framework Django, construído em Python, como a base para nosso desenvolvimento web. Esta escolha é motivada pelo fato de haver uma maior quantidade de pessoas no NTI com conhecimento em Django, o que facilitará a manutenção do sistema no futuro.

**Figma:** O Figma será utilizado para o design de interfaces de usuário (UI), permitindo a criação colaborativa de wireframes, protótipos e designs finais. Sua abordagem baseada em nuvem facilita a revisão e o compartilhamento de designs em tempo real.

**Visual Studio Code:** Este ambiente de desenvolvimento integrado (IDE) oferece uma ampla gama de recursos e extensões que aumentam a produtividade dos desenvolvedores. Sua integração perfeita com o Git e outras ferramentas torna-o uma escolha ideal para o desenvolvimento de software.

**13. Cronograma de Sprints**

| **Sprint** | **Data de Início** | **Data de Término** | **Tarefas** | **Entregas** |
| --- | --- | --- | --- | --- |
| 1 | 21/02/2024 | 27/02/2024 | Definir as regras de negócio com a Administração do RU, treinamento do sistema atualmente utilizado na impressão dos tickets com o NTI, projetar o modelo de dados, iniciar o desenvolvimento do CRUD | Modelo de dados |
| 2 | 28/03/2024 | 03/03/2024 | Finalizar o desenvolvimento do CRUD, testar o CRUD, documentar o CRUD, Planejar a interface do usuário | CRUD completo |
| 3 | 04/03/2024 | 29/04/2024 | Desenvolver a interface do usuário, integrar a interface do usuário com o backend, testar a interface do usuário, documentar a interface do usuário, planejar a autenticação do usuário | Interface do usuário |
| 4 | 30/04/2024 | 09/05/2024 | Desenvolver a autenticação do usuário, integrar a autenticação do usuário com o backend, testar a autenticação do usuário, documentar a autenticação do usuário, planejar a funcionalidade de compra de tickets | Autenticação do usuário |
|  | 10/05/2024 | 19/05/2024 | Desenvolver a funcionalidade de compra de tickets, integrar a funcionalidade de compra de tickets com o backend, testar a funcionalidade de compra de tickets, documentar a funcionalidade de compra de tickets, planejar a funcionalidade de validação de tickets | Compra de tickets |
| 5 | 20/05/2024 | 29/05/2024 | Desenvolver a funcionalidade de validação de tickets, integrar a funcionalidade de validação de tickets com o backend, testar a funcionalidade de validação de tickets, documentar a funcionalidade de validação de tickets, planejar a funcionalidade de relatórios | Validação de tickets |
| 6 | 30/05/2024 | 08/06/2024 | Desenvolver a funcionalidade de relatórios, integrar a funcionalidade de relatórios com o backend, testar a funcionalidade de relatórios, documentar a funcionalidade de relatórios, planejar a revisão final do projeto | Relatórios |
| 7 | 09/06/2024 | 18/06/2024 | Revisar todo o projeto, testar todo o projeto, documentar todo o projeto, preparar a apresentação final do projeto | Projeto finalizado |