**CONSTRUÇÃO DE COMANDAS DIGITAIS**

Danilo Sampaio Fonseca (1), Felippe da Silva Vieira (2), Luiz Gustavo Scottieri (3), Victor Teixeira Silva (4), Orientador: Prof°. Me. José Picovsky. ADS-RA 00341537 (1), ADS-RA 00343303 (2), ADS-RA 00341266 (3), ADS-RA 00342509 (4)

**RESUMO**

O tempo em que vivemos, pós pandemia de Covid-19 e, com as crises de produção e desemprego que a sucede, faz necessário levantar o debate sobre a migração de certos serviços ao mercado digital e a buscar por soluções que otimizem o tempo em que os usuários gastam, para ter acesso a certos produtos e serviços. Com esse pensamento que, o presente artigo, tem o objetivo de ilustrar as informações acerca do projeto Carambola, este que está sendo desenvolvido, com a ideia de agregar a logística de atendimento e melhorar a produtividade de estabelecimentos do ramo alimentício. A ideia é direcionar e, facilitar o acesso dos consumidores aos cardápios dos estabelecimentos, presentes no banco de dados da plataforma. Desse modo o desenvolvimento do projeto busca fornecer o acesso aos serviços do estabelecimento com melhor praticidade, melhorar a velocidade do atendimento e a facilitar o pagamento. [4]

**Palavras-Chave:** Aplicativo para Controle de Comandas Digitais; Projeto Carambola- Comandas Digitais; Aplicativo para Logística Bares.

**1.INTRODUÇÃO**

Com o passar dos anos, e seus eventos, o mercado de tecnologia vem sofrendo diversas mudanças. Um exemplo desses eventos foi a pandemia de Covid-19, que impactou diretamente na rotina de milhares de pessoas em todo o mundo, e essas mudanças na rotina afetaram consequentemente o mercado de trabalho.

Com as medidas sanitárias, restritivas de livre circulação nas ruas que tiveram de ser aplicadas pelas autoridades, durante esse período, as pessoas tiveram que se adaptar a migração de serviços convencionalmente físicos para as plataformas digitais. Como por exemplo a demanda crescente nos aplicativos de delivery, a virtualização dos serviços de agendamento de consultas médicas, a migração de cursos presenciais para aplicativos de educação EAD, entre inúmeros exemplos.

A princípio essas mudanças seriam apenas para o período referente a pandemia, mas atualmente elas estão se concretizando como rotina, por esses modelos de negócio se mostrarem eficientes ao atender as demandas da sociedade, principalmente em relação a otimização do tempo. E atualmente não é difícil se deparar com situações em que se faz necessário o uso de tecnologias que agilizam o e conectam clientes a produtos e serviços.

E dentro dessa corrente de mudança, no mercado de tecnologia que o Projeto Carambola tem a intenção de se estabelecer. A princípio como uma plataforma de pagamento para estabelecimentos do ramo de bebidas e alimentos, porém com a ideia de integrar a logística de atendimento, aumentando a produtividade dos funcionários transferindo as demandas para a aplicação, agilizando o tempo em que o cliente leva para escolher os pedidos, melhorando a fluidez no atendimento; o intuito é que esses serviços sejam aplicados em bares, restaurantes e baladas.

A implementação do sistema se daria por um aplicativo mobile hospedado nas lojas de aplicativos Android e IOS, facilitando o acesso a aplicação. Essa forma de distribuição da aplicação e ao invés de um sistema inteiro dedicado a um estabelecimento, torna economicamente viável para uma gama de empreendimentos tenham acesso a plataforma, pois haverá redução de gastos no quesito manutenção e implementação por parte do setor de TI.

**2. MÉTODOS**

O projeto Carambola, está em desenvolvimento, a etapa atual está pautada na construção de um Banco de dados relacional para ser utilizado junto a aplicação. A linguagem para a construção do banco de dados é a SQL (*Strutured Query Language*) utilizada no SGBD Oracle.

[1] Durante a década de 1970 se fez necessário a viabilização de uma linguagem que se adaptasse ao modelo teórico relacional, assim de um estudo de E.F. Codd, membro do laboratório de pesquisa da IBM em San Jose, Califórnia, surgiu o SQL. O primeiro modelo comercial com SQL surgiu no final da década de 1970, e sucesso da linguagem fez com que a ANSI (*American National Standarts Institute*) padronizasse a implementação do SQL. Em 1982 foi lançada a primeira versão padronizada da linguagem SQL, ganhando melhorias tornando-se a mais poderosa das ferramentas para manipulação de banco de dados até hoje, utilizada pela maioria dos bancos de dados existentes.

[2] Em 1977, foi fundada a Oracle por Larry Ellison, Bob Miner, Ed Oates e Bruce Scott, inicialmente chamada de Software Development Labs (SDL). Larry e Bob trabalharam juntos em um projeto de construção de um banco de dados para a CIA.

Lá desenvolveram um banco de dados relacional, curiosamente o nome do projeto era projeto Oracle. Em 1983 a Oracle lança seu primeiro RDBMS em 32 bits, escrito na linguagem C que ainda dava suporte ao anterior que foi escrito em Assembler.

Entre 1994 e 1995 morre Bob Miner e é lançado o primeiro banco de dados 64 bits. Em 1997 é lançado o Oracle 8. Junto com o Aplication server, o primeiro banco de dados para a web, e ferramentas para data warehousing, possuindo suporte nativo em Java.

Em 1998 o primeiro RDBMS para o Linux, e em 1999 o Oracle 8i integrando Java/XML nas ferramentas de desenvolvimento. Em 2003 o Oracle 10g é lançado com banco de dados voltado a grandes servidores, e é chamado de Grid de servidores.

[3] Entre 2007 e 2009 veio a grande expansão da empresa com o lançamento do Oracle 11g e com a compra da Sun Microsystems, que era dona do Java e também do código fonte de MySQL.

Em 2013 veio a versão 12c, que teve grande aceitação entre as empresas, considerada uma das melhores até hoje, depois de quase 10 anos de seu lançamento ainda está em produção em muitos projetos. Em 2018 vieram as versões 18c e 19c, e atualmente a Oracle se encontra na versão 21c.

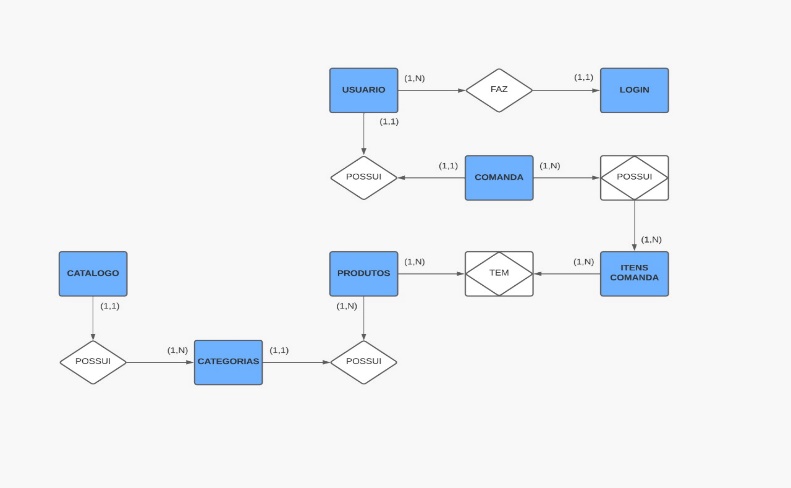
**3. DESENVOLVIMENTO**

No início, o desenvolvimento do banco de dados foi através de um DER (Diagrama de entidade relacionamento) como mostrado na figura 1, utilizado para descrever os objetos do mundo real através de entidades e seus relacionamentos.

Para demostrar a utilização do DER no projeto podemos pegar como exemplo a entidade COMANDA, que se relaciona com a entidade ITENS COMANDA através da entidade associativa POSSUI. Entidade essa, que também pode ser chamada de entidade fraca, sua principal característica é ser portadora das chaves primarias, das duas entidades que se relacionam através dela.

No modelo conceitual é possível que vejamos a cardinalidade de relacionamento entre as entidades, repare que na figura 2 o relacionamento entre ITENS COMANDA e PRODUTOS possuem cardinalidade (1,N), isso é, muitos para muitos, isto nos sugere manter os atributos desse relacionamento contidos dentro de uma entidade relacional, parece simples, porém é de grande auxílio no momento de passar o modelo conceitual para o modelo lógico.

**Figura 1 – DER (Diagrama de Entidade Relacionamento)**

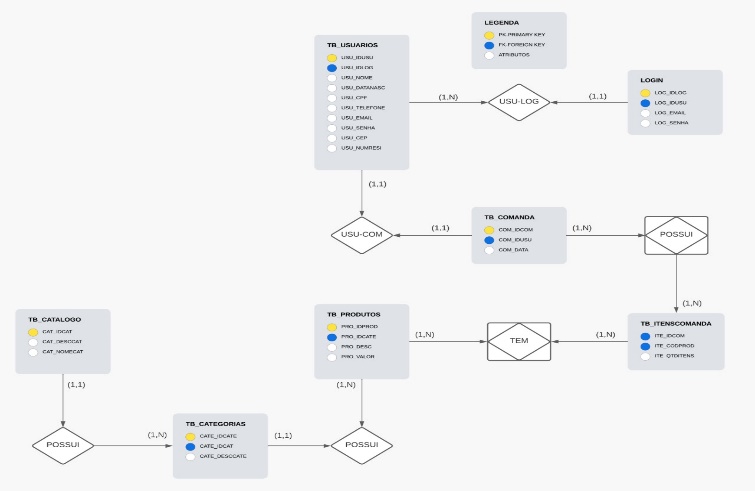
****

**Fonte:** Victor Teixeira Silva, Danilo Sampaio Fonseca, Felippe da Silva Vieira, Luiz Gustavo Scottieri.

Após o desenvolvimento do modelo conceitual, o projeto seguiu com o modelo o lógico. Esse modelo se trata do MER (Modelo de Entidade Relacionamento) momento em que são definidas as tabelas, os campos, atributos comuns e identificadores como chaves primarias e constraints como veremos futuramente nas figuras 2 e 3, que irão compor a estrutura do banco de dados da aplicação, é o principal modelo usado em diversos projetos, pois facilita muito o entendimento de como ficará a estrutura do banco de dados.

Uma demonstração prática de como o modelo lógico do projeto foi desenvolvido, pode ser visto na figura 2. La estão as entidades TB\_USUARIOS e LOGIN, que se relacionam através USU-LOG, contando com as chaves primarias USU\_IDUSU, LOG\_IDLOG e as chaves estrangeiras USU\_IDLOG, LOG\_IDUSU atributos esses que fazem o relacionamento entre as duas entidades.

**Figura 2 - MER (Modelo de Entidade Relacionamento)**

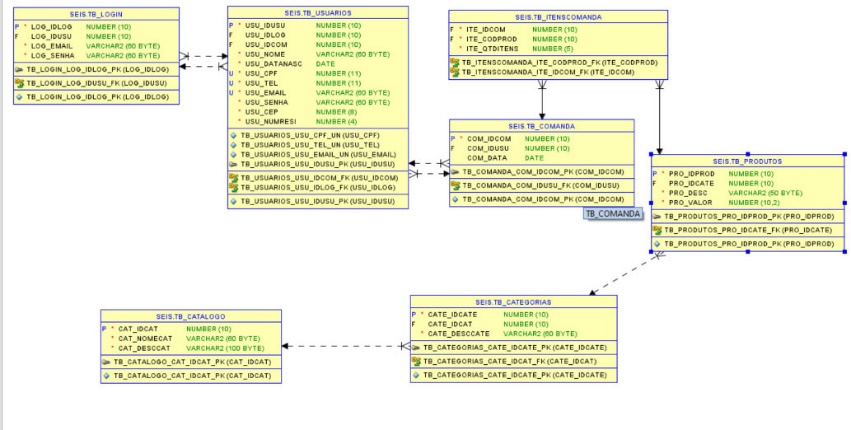
****

**Fonte:** Victor Teixeira Silva, Danilo Sampaio Fonseca, Felippe da Silva Vieira, Luiz Gustavo Scottieri.

Outro exemplo pode ser visto na figura 3, imagem que descreve o modelo gerado pelo SGBD *SQL Developer*. Nele é possível visualizar o relacionamento entre as tabelas SEIS.TB\_COMANDA e SEIS.TB\_PRODUTOS, relacionamento que acontece através da entidade associativa SEIS.TB\_ITENSCOMANDA e dos atributos identificadores, chaves primarias das tabelas anteriores, entretanto chaves estrangeiras nessa tabela, ITE\_IDCOM e a ITE\_IDCODPROD.

É possível observar essas respectivas chaves estrangeiras, são definidas e referenciadas, nas constraints da tabela SEIS.TB\_ITENSCOMANDA. Outro relacionamento que podemos acompanhar através da imagem, é o que acontece entre a SEIS.TB\_USUARIOS e a SEIS.TB\_COMANDA relacionamento que ocorre através das chaves primárias USUS\_IDUSU, COM\_IDCOM e as chaves estrangeiras USU\_IDCOM, COM\_IDUSU.

**Figura 3 - MER (Modelo de Entidade Relacionamento)**

****

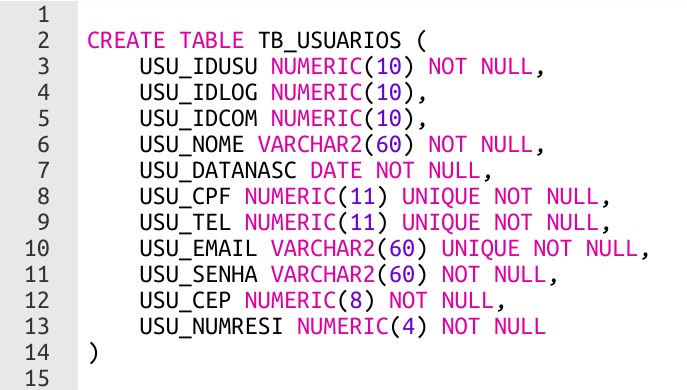
**Fonte:** Victor Teixeira Silva, Danilo Sampaio Fonseca, Felippe da Silva Vieira, Luiz Gustavo Scottieri.

Após a o desenvolvimento do modelo lógico começa-se a ser desenvolvido o Script do banco de dados, dentro do *SQL Developer*. Na figura 4 estão presentes exemplos, dos comandos CREATE TABLE na criação das tabelas do banco de dados do projeto.

Note que o comando CREATE TABLE TB\_USUARIOS, é o que dá início a formação de uma tabela para serem criadas as colunas, já o comando USU\_CPF NUMERIC (11) UNIQUE NOT NULL define o nome da coluna, atenção a definição de variável NUMERIC (11) que informa o tipo da variável a ser armazenadas nas linhas dessa coluna, no caso do exemplo temos um tipo numeric que indica que ali deve ser armazenado apenas números de precisão exata.

Outra parte do exemplo em questão, são os comandos UNIQUE, que indica que os dados armazenados naquela linha não podem se repetir em outra linha daquela coluna, e o comando NOT NULL que indica a que a aquela linha não deverá ficar em branco, obrigatoriedade na inscrição de dados.

**Figura 4 - Exemplos do Comando CREATE TABLE**

****

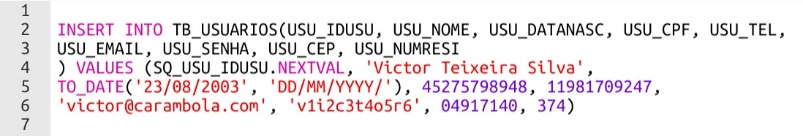
**Fonte:** Victor Teixeira Silva, Danilo Sampaio Fonseca, Felippe da Silva Vieira, Luiz Gustavo Scottieri.

Seguindo com a demonstração do Script do projeto, passamos agora para a etapa de inserir dados nas tabelas.

E os comandos exemplificados na figura 5 são os INSERT INTO, que direciona e insere os valores na respectiva coluna e o comando VALUE, que define esses valores de acordo com a ordem pré-estabelecida no INSERT INTO e o tipo de dados no CREATE TABLE.

Abaixo na figura 5 há um exemplo do comando INSERT INTO na tabela referente aos dados dos usuários do projeto, seguido pela ordem das colunas, em que os valores serão inseridos nessa tabela.

**Figura 5 – Exemplos do Comando INSERT INTO**

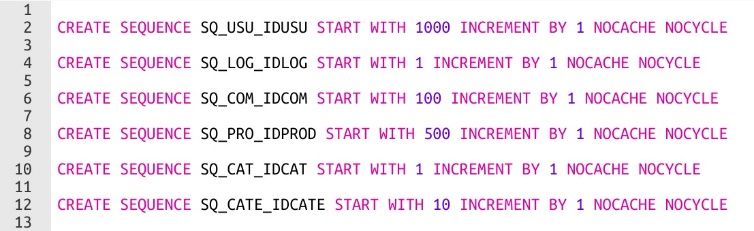
****

**Fonte:** Victor Teixeira Silva, Danilo Sampaio Fonseca, Felippe da Silva Vieira, Luiz Gustavo Scottieri.

É importante ressaltar a função SQ\_USU\_IDUSU.NEXTVAL, que deriva do comando SEQUENCE presente na figura 6, utilizado pra criar valores numéricos sequenciais que são gerados automaticamente pelo Oracle.

Outra função de grande importância é a TO\_DATE usada para converter strings, em tipos de dados Data/Hora.

**Figura 6 – Exemplos do comando SEQUENCE**

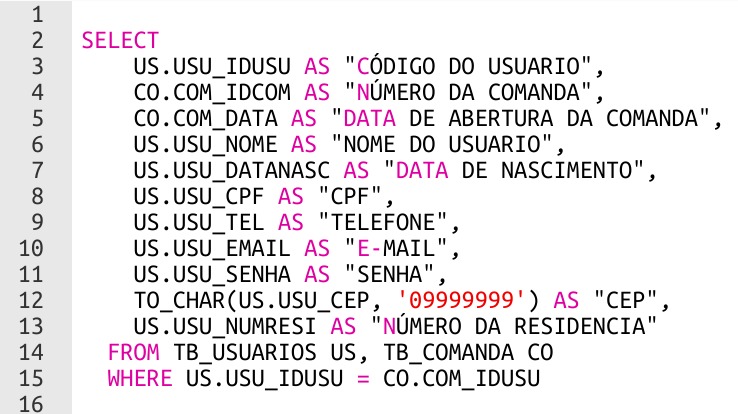
****

**Fonte:** Victor Teixeira Silva, Danilo Sampaio Fonseca, Felippe da Silva Vieira, Luiz Gustavo Scottieri.

Com as tabelas criadas e valores inseridos, se faz necessário entender como consultar esses valores? Para isso usaremos o comando SELECT.

Na figura 7 temos um exemplo desse comando aplicado na TB\_USUARIOS, usando o comando FROM para unir a pesquisa e relacionar os dados com os da TB\_COMANDA. Importante ressaltar a utilização do comando AS dentro da sintaxe do SELECT, que define um apelido para a consulta.

**Figura 7 – Exemplo do comando SELECT na tabela usuários.**

****

**Fonte:** Victor Teixeira Silva, Danilo Sampaio Fonseca, Felippe da Silva Vieira, Luiz Gustavo Scottieri.

O resultado da consulta referindo as tabelas citadas no parágrafo anterior pode ser visto na Figura 8.

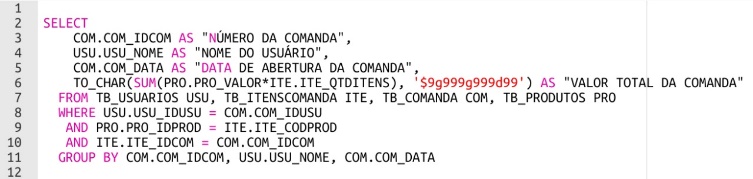
**Figura 8 – Resultado da consulta na tabela usuários.**

****

**Fonte:** Victor Teixeira Silva, Danilo Sampaio Fonseca, Felippe da Silva Vieira, Luiz Gustavo Scottieri.

Neste exemplo temos, a consulta retornando a soma do valor total do que foi consumido pelo cliente. Para isso utiliza-se primeiramente o comando SUN para obter a soma do valor numérico e logo em seguida o comando TO\_CHAR para converter a soma dos valores em uma string informando o preço. A figura 9 mostra o a sintaxe da pesquisa com apenas quatro colunas selecionadas pelo FROM, cada coluna referente a uma tabela diferente, com as regras de pesquisa definidas pelo WHERE para evitar duplicidade de dados, algo que pode ocasionar um erro na consulta.

**Figura 9 -Consulta com o SELECT usando múltiplas colunas na tabela comanda.**

****

**Fonte:** Victor Teixeira Silva, Danilo Sampaio Fonseca, Felippe da Silva Vieira, Luiz Gustavo Scottieri.

E por fim os dados são agrupados pelo comando GROUP BY gerando o resultado que pode ser visualizado na figura 10.

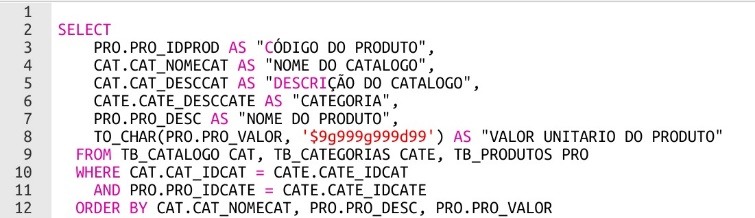
**Figura 10 – Resultado da consulta na tabela comanda.**

****

**Fonte:** Victor Teixeira Silva, Danilo Sampaio Fonseca, Felippe da Silva Vieira, Luiz Gustavo Scottieri.

Na figura 11 contém outro exemplo da utilização do SELECT consultas e agrupamento de tabelas, se trata de uma junção das TB\_CATALOGO, TB\_CATEGOTIA e TB\_PRODUTOS, nesse caso a consulta organiza de forma hierárquica, onde cada estabelecimento com um catalogo único, com categoria e essas categorias com diversos produtos, organizando os estabelecimentos por linhas.

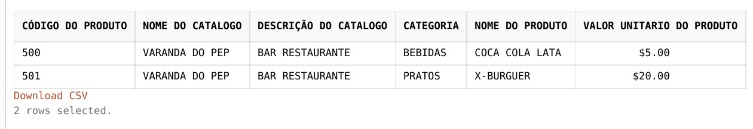
**Figura 11 – Consulta com o SELECT usando múltiplas tabelas para produtos.**

****

**Fonte:** Victor Teixeira Silva, Danilo Sampaio Fonseca, Felippe da Silva Vieira, Luiz Gustavo Scottieri.

O resultado da consulta realizada no parágrafo anterior, pode ser visualizado na Figura 12.

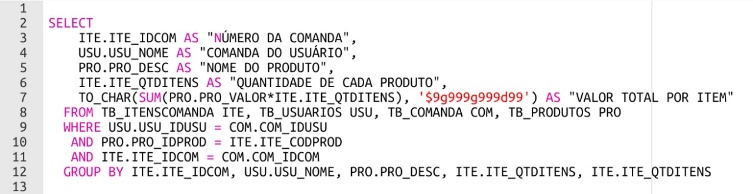
**Figura 12 – Resultado da consulta das tabelas catálogos, categorias, produtos.**

****

**Fonte:** Victor Teixeira Silva, Danilo Sampaio Fonseca, Felippe da Silva Vieira, Luiz Gustavo Scottieri.

A figura 13 apresenta a sintaxe para a junção de quatro tabelas, próxima do que que foi apresentado nas consultas anteriores. Entretanto o objetivo nessa consulta é mostrar item por item, junto ao número da comanda, ao preço de cada item, nome do usuário titular da comanda e a quantidade de cada item presentes nela.

**Figura 13 - Consulta com uso do SELECT nas tabelas Usuários, Itens comanda, comanda e Produtos.**

****

**Fonte:** Victor Teixeira Silva, Danilo Sampaio Fonseca, Felippe da Silva Vieira, Luiz Gustavo Scottieri.

Na figura 14, contém o resultado da consulta anterior.

**Figura 14 – Resultado da consulta nas tabelas da figura 13.**

****

**Fonte:** Victor Teixeira Silva, Danilo Sampaio Fonseca, Felippe da Silva Vieira, Luiz Gustavo Scottieri.

**4. CONSIDERAÇÕES FINAIS**

Pode-se concluir que com a aplicação ainda em desenvolvimento, por etapas, que estão sendo concluídas ao longo dos semestres do curso, ainda faltam muitos detalhes do projeto a serem desenvolvidos pelo grupo. Entretanto em relação a modelagem de dados e a estruturação de um banco de dados relacional, neste artigo estão presentes resultados satisfatórios.

Os objetivos desta etapa se resumiram a ilustrar de forma abstrata, usando o modelo conceitual, os relacionamentos entre usuários e estabelecimentos através da ótica do banco de dados. Mostrar de forma aprofundada as tabelas e atributos, utilizando o modelo lógico.

E através da criação do modelo físico, apresentar as sintaxes de construção, de inserção e de consultas de dados que foram utilizadas no projeto, demonstrar essas consultas através da cadeia de comandos presentes no SELECT e apresentar o resultado dessas consultas, nas tabelas, através das imagens anexadas no artigo.

Futuramente é pretendido que seja finalizado a versão Mobile do sistema e também a versão web.

Junto disso se inicia o desenvolvimento de uma versão exclusiva para os estabelecimentos, obterem acesso a controle de estoque, relatórios de vendas e impulsionar por meio de trafego pago seu próprio estabelecimento.

**5. REFERÊNCIAS**

[1] Devmedia. **Entendendo o SQL**. Disponível em: <https://www.devmedia.com.br/entedendo-a-linguagem-sql/7775>

[2] Devmedia. **História da Oracle.** Disponível em: https://www.devmedia.com.br/historia-da-oracle/4685

[3] Oracle. **Sobre a Oracle, Uma História de possibilidades.** Disponível em: https://www.oracle.com/br/corporate/

[4] Decision. **Pesquisa aponta que brasileiro ampliou uso de pagamentos digitais.** Disponível em: https://www.decisionreport.com.br/financas/brasileiro-amplia-uso-pagamentos-digitais-aponta-pesquisa/#.Y1LKRuTMK3A