

UNIVERSIDADE DO VALE DO ITAJAÍ - UNIVALI
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO
DE SISTEMAS

HANDS ON WORKS VI

PAULO IGOR LOPES
paulo.lobes@edu.univali.br

APP CONDOMÍNIO

FLORIANÓPOLIS, JUNHO de 2023

Resumo:

Este trabalho tem como objetivo criar um aplicativo android com gerenciamento de banco de dados MySql. O mesmo irá implementar um sistema de gerenciamento de condomínio o conceito CRUD (Create, Read, Update, Delete), classificando-o com o um produto de informática, que será usado normalmente em aparelho Android.

Introdução:

Este trabalho foi desenvolvido criando um sistema de gerenciamento de condomínio, a fim de facilitar a vida do síndico e resguardando informações e prevalecer os dados. O banco de dados utilizado foi o MySQL

.

APP CONDOMÍNIO

Este app tem como objetivo facilitar a vida do síndico do Condomínio Villa do Mar, catalogando os apartamentos e informando o nome dos respectivos moradores/proprietários e um campo de Status mostrando a situação em que se encontra cada apartamento. Também inclui a função de reservas do salão de festas.

Com estas informações será fácil de identificar futuras datas disponíveis para utilizar o salão de festas, verificar qual apartamento está vago, qual está em reformas e quais estão ocupados.

Recursos Necessários e Ciclo de Vida:

Fase 1: Prototipação e estudo da ferramenta e linguagem escolhida.

Recursos:

1 Programador/ Analista de Software

1 Computador com acesso à internet

Fase 2: Codificação e testes iniciais.

Recursos:

1 Programador/Analista de Software/Tester

1 Computador com acesso à internet

1 Smartphone Android

Fase 3: Implementar versão 1.0 e liberar a utilização do software

Plano de Testes:

1. Objetivos dos testes:

- Verificar a funcionalidade correta do aplicativo em relação ao gerenciamento de apartamentos.
- Garantir a integridade dos dados dos apartamentos e dos moradores.
- Validar a capacidade do aplicativo de atualizar o status dos apartamentos corretamente.

2. Escopo dos testes:

- Testar as principais funcionalidades do aplicativo relacionadas ao gerenciamento de apartamentos.
- Não incluir testes de recursos adicionais não mencionados, como pagamentos de condomínio, reservas de áreas comuns, etc.

3. Estratégia de testes:

- Realizar testes manuais, utilizando uma abordagem de teste exploratório para interagir com o aplicativo.
- Validar as diferentes ações possíveis, como adicionar um novo apartamento, editar informações, atualizar o status, etc.

4. Casos de teste:

a) Adicionar novo apartamento

- Descrição: Verificar se é possível adicionar um novo apartamento ao sistema.
- Passos:
 1. Acessar a funcionalidade de adicionar novo apartamento.
 2. Preencher os campos obrigatórios, como número do apartamento, nome do morador e status.
 3. Salvar as informações.
- Resultados esperados:
 1. O apartamento é adicionado com sucesso e aparece na busca quando inserido o seu número para pesquisa.

b) Atualizar status do apartamento

- Descrição: Verificar se é possível atualizar o status de um apartamento existente.
- Pré-condições: Um apartamento existente no sistema.
- Passos:
 1. Localizar o apartamento desejado.
 2. Selecionar a opção de atualizar o status.
 3. Inserir um novo Status (sob reformas, habitado, vazio).
 4. Salvar as alterações.
- Resultados esperados:
 1. O status do apartamento é atualizado com sucesso e aparece na busca quando inserido o seu número para pesquisa.

5. Critérios de entrada/saída:

- Os casos de teste devem ser executados sem erros ou falhas.
- Os resultados esperados devem ser alcançados de acordo com as descrições dos casos de teste.

6. Recursos necessários:

- Dispositivos móveis ou emuladores adequados para testar o aplicativo.
- Ambiente de teste com acesso às funcionalidades do aplicativo.

7. Riscos e mitigação:

- Possíveis riscos incluem falhas na atualização do status do apartamento ou perda de dados.
- Para mitigar esses riscos, faça backup regular dos dados do aplicativo e teste cuidadosamente as atualizações de status.

Análise de Riscos:

1. Identificação de riscos:

- Falta de experiência da equipe com a tecnologia utilizada:
 - Gatilho: Quando um membro da equipe não estiver familiarizado com a tecnologia utilizada, deve-se atribuir um mentor mais experiente para ajudá-lo.
 - Plano de contingência: Se a falta de experiência da equipe resultar em um erro ou problema significativo, o mentor deve ser contatado imediatamente para ajudar a resolver o problema e, posteriormente, oferecer treinamento adicional para a equipe.
- Mudanças nos requisitos do cliente:
 - Gatilho: Quando houver uma mudança significativa nos requisitos do cliente, o gerente do projeto deve ser notificado imediatamente.
 - Plano de contingência: O gerente do projeto deve avaliar os impactos da mudança nos requisitos e, se necessário, reunir a equipe para discutir a nova estratégia e ajustar o cronograma do projeto.
- Falhas na segurança do banco de dados:
 - Gatilho: Quando ocorrer uma falha na segurança do banco de dados, uma notificação automática deve ser enviada para o administrador do banco de dados.
 - Plano de contingência: O administrador do banco de dados deve tomar medidas imediatas para proteger os dados e corrigir a falha, enquanto a equipe de desenvolvimento deve avaliar a causa raiz do problema e implementar medidas de segurança

DICIONÁRIO DE DADOS

Título: APP CONDOMÍNIO

Descrição: sistema de gerenciamento de Apartamentos do condomínio.

Tabela: Condomínio			
#	Nome	Tipo	Agrupamento
1	AP	int(3)	
2	Prorietario	varchar(100)	utf8_general_ci
3	Situacao	varchar(50)	utf8_general_ci

Descrição: sistema de gerenciamento de Reserva do Salão de Festas do condomínio.

#	Campo	Tipo	Agrupamento
1	ID_Reserva	int	
2	Data	date	
3	Horario_Inicio	time	
4	Horario_Termino	time	
5	Nome_Morador	varchar(100)	utf8_general_ci
6	Numero_Apartamento	int	
7	Observacoes	text	utf8_general_ci

Código SQL:

```
CREATE DATABASE condominio;
```

```
USE condominio;
```

```
CREATE TABLE apartamentos (  
  Ap INT PRIMARY KEY,  
  Proprietario VARCHAR(50) NOT NULL,  
  Situacao VARCHAR(50) NOT NULL,
```

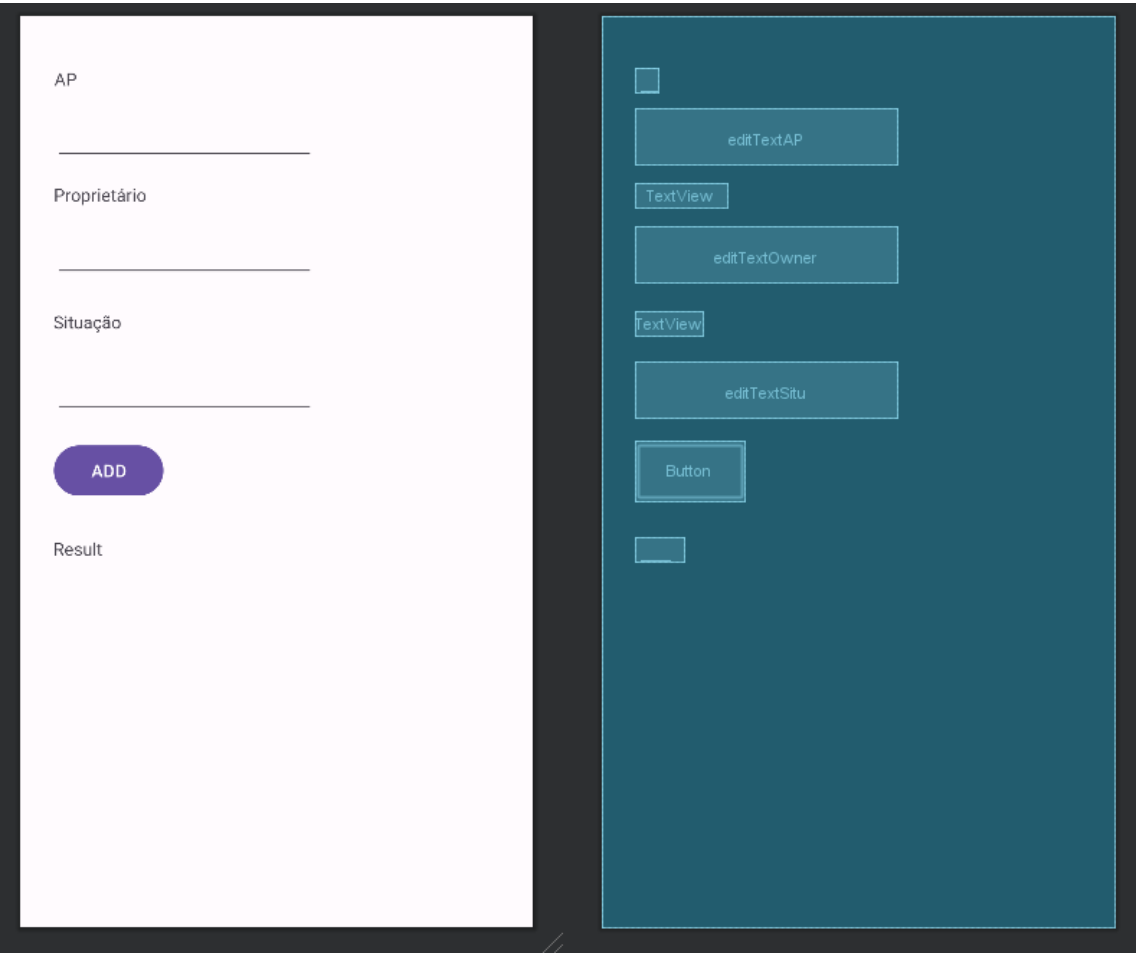
```
CREATE TABLE ReservaSalaoFestas (  
  ID_Reserva INT,  
  Data DATE,  
  Horario_Inicio TIME,  
  Horario_Termino TIME,  
  Nome_Morador VARCHAR(100) COLLATE utf8_general_ci,  
  Numero_Apartamento INT,  
  Observacoes TEXT COLLATE utf8_general_ci  
);
```

Ao iniciar o App '**APP CONDOMÍNIO**', o usuário se depara com a tela de inicialização. Clicando em ENTRAR ele será levado para a próxima tela. Na segunda tela, é possível selecionar as seguintes opções:

Consultar Apartamentos: Nesta opção, o usuário poderá pesquisar qualquer um dos apartamentos do condomínio pelo seu número, adicionar ou remover informações como o nome do morador/proprietário e a situação do apartamento, declarando um status de acordo com o que está acontecendo no momento naquele apartamento em específico.

Reservas do Salão de Festas: Nesta opção, o síndico poderá fazer reservas para os condôminos que desejem usar o salão de festas em alguma data específica.

Prototipação e Interface:



Referencias:

MICROSOFT. **Documentação de SQL.** 2023. Disponível em: <https://learn.microsoft.com/pt-br/sql/relational-databases/performance/joins?view=sql-server-ver16>. Acesso em: 15 junho. 2023.

CRIANDOBITS. **Formatação de valores e datas para inserção em banco de dados.** 2014. Disponível em: <https://www.youtube.com/watch?v=rljIJvQ3hUs>. Acesso em: 17 junho. 2023.

MICROSOFT. **Documentação .Net SQL Data Reader:** métodos. Métodos. Assembly: System.Data.SqlClient.dll. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/api/system.data.sqlclient.sqldatareader.getdatetime?view=dotnet-plat-ext-5.0>. Acesso em: 17 junho. 2023.