

Computação II — Atividade de Extensão (2026.1)

Mapa Inteligente de Valores Imobiliários

Desenvolvimento de ferramenta web para análise territorial, mercado imobiliário local e qualidade de dados

Resumo da proposta

A turma irá construir, em grupos, uma aplicação web simples usando HTML, CSS e JavaScript puro para visualizar dados imobiliários de Matinhos/PR em mapas, gráficos, indicadores e tabelas de qualidade da base. A atividade será desenvolvida como extensão, com possibilidade de apresentação para imobiliárias, corretores, gestores públicos ou atores locais interessados no território.

Tecnologias: HTML, CSS, JavaScript, JSON, Leaflet, Chart.js e IA como apoio orientado

1. Apresentação da atividade

A atividade propõe o desenvolvimento colaborativo de uma ferramenta web para visualizar e analisar dados imobiliários da cidade de Matinhos/PR. O sistema deverá transformar uma base tabular em um painel digital com mapas, gráficos, indicadores e diagnóstico de qualidade dos dados.

O objetivo é aproximar os conteúdos de Computação II de uma demanda real: compreender o território, organizar dados e criar uma solução que possa ser demonstrada para imobiliárias, corretores, gestores públicos ou instituições relacionadas à análise patrimonial.

Os alunos não precisam iniciar a atividade sabendo programar tudo. O projeto será desenvolvido por etapas, com código simples, uso de exemplos, orientação em sala e apoio controlado de ferramentas de Inteligência Artificial.

2. Enquadramento extensionista

A atividade é extensionista porque conecta a disciplina a um problema territorial real e a potenciais usuários externos. A ferramenta desenvolvida poderá auxiliar a leitura de informações imobiliárias, identificar regiões com diferentes faixas de valor e revelar problemas de qualidade na base de dados.

O público externo pode incluir imobiliárias locais, corretores de imóveis, gestores públicos, equipes técnicas de projetos de avaliação imobiliária, instituições de pesquisa e comunidade interessada no mercado imobiliário de Matinhos e litoral do Paraná.

3. Objetivos da atividade

3.1 Objetivo geral

Desenvolver, de forma colaborativa, uma aplicação web para visualização e análise de dados imobiliários, utilizando HTML, CSS, JavaScript, dados em JSON, mapas interativos, gráficos e apoio orientado de ferramentas de Inteligência Artificial.

3.2 Objetivos específicos

- Entender a estrutura básica de uma aplicação web com HTML, CSS e JavaScript.
- Ler e interpretar dados organizados em JSON.
- Criar elementos visuais em uma página web, como cards, tabelas e seções.
- Exibir dados em tabelas, indicadores e gráficos.
- Plotar informações em mapa interativo usando Leaflet.
- Criar filtros simples por bairro, logradouro ou faixa de valor.
- Identificar problemas de qualidade em uma base de dados.
- Usar IA como apoio à programação, revisão e explicação de código.
- Documentar o que foi desenvolvido.
- Apresentar o produto final para um público externo ou banca simulada.

4. Produto final esperado

Ao final da atividade, a turma deverá entregar um painel web com quatro módulos principais. Cada módulo será desenvolvido por um grupo, mas todos deverão ser integrados em uma única aplicação.

Módulo	Descrição
Mapa Geral	Exibe imóveis no mapa quando houver latitude e

	longitude. Cada marcador deve apresentar um popup com dados básicos do imóvel.
Mapa de Valor	Classifica imóveis por faixas de valor do m ² de testada, com legenda.
Dashboard	Apresenta indicadores, gráficos e rankings, como total de registros, área total, valor médio por bairro e logradouros com maior valor.
Qualidade da Base	Identifica problemas nos dados, como ausência de coordenadas, CEP, bairro, valor, área ou data de avaliação.

5. Organização dos alunos

A turma será organizada em quatro grupos. Cada grupo terá uma missão técnica específica e deverá colaborar com os demais na integração final.

Grupo 1 — Mapa Geral

- Carregar os dados para uso no mapa.
- Verificar quais registros possuem latitude e longitude.
- Criar o mapa base com Leaflet.
- Adicionar marcadores e popups.
- Exibir quantidade de registros com e sem coordenadas.

Grupo 2 — Mapa de Valor

- Criar faixas de valor do m² de testada.
- Colorir os pontos conforme a faixa de valor.
- Criar legenda visual.
- Atualizar o mapa conforme os filtros.

Grupo 3 — Dashboard

- Criar cards de indicadores.
- Calcular totais, médias e rankings.
- Gerar gráficos com Chart.js.
- Criar ranking de bairros ou logradouros.
- Explicar o significado dos indicadores.

Grupo 4 — Qualidade da Base

- Identificar dados ausentes ou inconsistentes.
- Criar indicadores de qualidade.
- Montar tabela de problemas.
- Calcular score de qualidade.
- Sugerir melhorias futuras para a base.

6. Estrutura técnica do projeto

O projeto deverá ser simples, sem backend e sem banco de dados. A aplicação será estática e poderá ser aberta com o Live Server do VS Code ou Google Colab.

```
projeto-imobiliario/  
|  
|-- index.html  
|  
|-- data/  
|   |-- matinhos_pgv.json  
|  
|-- assets/  
|   |-- css/  
|       |-- style.css  
|   |  
|   |-- js/  
|       |-- app.js  
|       |-- maps.js  
|       |-- dashboard.js  
|       |-- quality.js  
|       |-- calculations.js  
|       |-- formatters.js
```

O arquivo JSON será a base de dados da aplicação. Ele deve conter uma lista de imóveis com campos como bairro, endereço, valor_m2_testada, area_terreno, valor_patrimonial, latitude e longitude.

7. Estratégia de desenvolvimento em camadas

Para reduzir a dificuldade, o projeto será desenvolvido em camadas. Cada grupo deve começar por uma versão mínima funcionando antes de avançar para recursos mais sofisticados.

Camada 1 — Visualização simples

Fazer aparecer algo na tela: total de registros, uma tabela simples ou um card.

Camada 2 — Cálculos

Transformar dados em informação: somas, médias, contagens, agrupamentos e rankings.

Camada 3 — Gráficos

Transformar informação em visual: gráficos de barras, pizza, rankings e distribuição de valores.

Camada 4 — Mapas

Visualizar espacialmente: mapa base, marcadores, popups, cores por faixa de valor e filtros.

Regra de ouro

Primeiro faça funcionar de forma simples. Depois melhore o visual, adicione filtros, gráficos ou novas funcionalidades. Um recurso simples funcionando vale mais do que uma tela avançada quebrada.

8. Cronograma sugerido

Encontro	Tema	Objetivo	Entrega
1	Apresentação do problema e da atividade	Entender o projeto, o público externo, os módulos e a divisão dos grupos.	Descrição curta do módulo do grupo.

2	Entendendo os dados	Abrir o JSON, identificar campos e compreender os tipos de dados.	Tabela com campo usado, finalidade e tipo de dado.
3	HTML, CSS e exibição básica	Exibir informações simples na tela usando JavaScript.	Um card, tabela ou informação funcionando.
4	Funções, condições e cálculos	Criar funções com if, forEach, acumuladores, médias e contagens.	Três cálculos funcionando.
5	Gráficos	Criar visualizações com Chart.js.	Um gráfico ou ranking funcionando.
6	Mapas com Leaflet	Criar mapa, marcadores, popups e validação de coordenadas.	Mapa carregando ou aviso adequado de ausência de coordenadas.
7	Integração e acabamento	Juntar os módulos, revisar erros, padronizar visual e testar.	Versão integrada do painel.
8	Apresentação extensionista	Apresentar o produto como demonstração para público externo ou banca simulada.	Apresentação final de 5 minutos por grupo.

9. Roteiro prático para os grupos

Passo 1 — Entenda seu módulo

- O que minha tela precisa mostrar?
- Quais dados preciso usar?
- Qual cálculo preciso fazer?
- Qual será o resultado visual?

Passo 2 — Localize os campos no JSON

- bairro
- endereco
- valor_m2_testada
- area_terreno
- valor_patrimonial
- latitude
- longitude

Passo 3 — Faça aparecer algo simples

- Antes de tentar gráfico ou mapa, mostre um texto, um número ou uma tabela simples na tela.

Passo 4 — Faça um cálculo

- Use forEach, if e acumuladores para calcular totais, médias ou contagens.

Passo 5 — Transforme o cálculo em função

- Organize o código em funções simples, com nomes claros.

Passo 6 — Mostre o resultado na tela

- Use document.getElementById e innerText ou innerHTML.

Passo 7 — Melhore com gráfico, tabela ou mapa

- Somente depois que o cálculo funcionar, transforme o resultado em uma visualização.

11. Dicas de código para alunos iniciantes

11.1 Mostrar total de registros

```
function contarRegistros(dados) {  
  return dados.length;  
}  
  
function mostrarTotalRegistros(dados) {  
  const total = contarRegistros(dados);  
  document.getElementById("total-registros").innerText = total;  
}
```

11.2 Somar área total dos terrenos

```
function calcularAreaTotal(dados) {  
  let total = 0;  
  
  dados.forEach(item => {  
    if (item.area_terreno) {  
      total = total + item.area_terreno;  
    }  
  });  
  
  return total;  
}
```

11.3 Calcular valor médio do m² de testada

```
function calcularMediaValorM2(dados) {  
  let soma = 0;  
  let quantidade = 0;  
  
  dados.forEach(item => {  
    if (item.valor_m2_testada && item.valor_m2_testada > 0) {  
      soma = soma + item.valor_m2_testada;  
      quantidade = quantidade + 1;  
    }  
  });  
  
  if (quantidade === 0) {  
    return 0;  
  }  
  
  return soma / quantidade;  
}
```

11.4 Contar registros sem coordenadas

```
function contarSemCoordenadas(dados) {  
  let total = 0;  
  
  dados.forEach(item => {  
    if (!item.latitude || !item.longitude) {  
      total = total + 1;  
    }  
  });  
  
  return total;  
}
```

11.5 Agrupar valor médio por bairro

```
function calcularMediaPorBairro(dados) {  
  const grupos = {};  
  
  dados.forEach(item => {  
    const bairro = item.bairro || "Não informado";
```

```
const valor = item.valor_m2_testada || 0;

if (!grupos[bairro]) {
  grupos[bairro] = { soma: 0, quantidade: 0 };
}

if (valor > 0) {
  grupos[bairro].soma = grupos[bairro].soma + valor;
  grupos[bairro].quantidade = grupos[bairro].quantidade + 1;
}
});

const resultado = [];

Object.keys(grupos).forEach(bairro => {
  const grupo = grupos[bairro];
  const media = grupo.quantidade > 0 ? grupo.soma / grupo.quantidade : 0;

  resultado.push({
    bairro: bairro,
    media: media
  });
});

return resultado;
}
```

11.6 Classificar faixas de valor

```
function classificarFaixaValor(valor) {
  if (!valor || valor <= 0) {
    return "Sem valor";
  }

  if (valor <= 100) {
    return "Até R$ 100/m²";
  }

  if (valor <= 200) {
    return "R$ 100 a R$ 200/m²";
  }

  if (valor <= 300) {
    return "R$ 200 a R$ 300/m²";
  }

  if (valor <= 500) {
    return "R$ 300 a R$ 500/m²";
  }

  return "Acima de R$ 500/m²";
}
```

11.7 Exemplo simples de marcador no Leaflet

```
function adicionarMarcadoresNoMapa(mapa, dados) {
  dados.forEach(item => {
    if (item.latitude && item.longitude) {
      L.marker([item.latitude, item.longitude])
        .addTo(mapa)
        .bindPopup(`
          <strong>${item.endereco}</strong><br>
          Bairro: ${item.bairro || "Não informado"}<br>
          Valor m²: R$ ${item.valor_m2_testada || 0}
        `);
    }
  });
}
```

11.8 Exemplo simples de gráfico com Chart.js

```
function criarGraficoBairros(labels, valores) {
  const contexto = document.getElementById("grafico-bairros");

  new Chart(contexto, {
    type: "bar",
    data: {
      labels: labels,
      datasets: [{
        label: "Valor médio m² testada",
        data: valores
      }]
    },
    options: {
      responsive: true,
      plugins: {
        legend: {
          display: true
        }
      }
    }
  });
}
```

11.9 Calcular score de qualidade de um registro

```
function calcularScoreRegistro(item) {
  const campos = [
    "rip_imovel",
    "municipio",
    "bairro",
    "endereco",
    "cep",
    "latitude",
    "longitude",
    "valor_m2_testada",
    "area_terreno",
    "valor_patrimonial",
  ];
```

```
"data_avaliacao_imovel"  
];  
  
let preenchidos = 0;  
  
campos.forEach(campo => {  
  if (item[campo]) {  
    preenchidos = preenchidos + 1;  
  }  
});  
  
return (preenchidos / campos.length) * 100;  
}
```

12. Missões detalhadas por grupo

Grupo 1 — Mapa Geral

Criar uma tela que mostre os imóveis no mapa.

Entrega mínima

- Mapa carregando
- Contagem de imóveis com coordenadas
- Contagem de imóveis sem coordenadas
- Marcadores no mapa
- Popup com dados do imóvel

Perguntas que o grupo deve saber responder

- O que é latitude?
- O que é longitude?
- Por que alguns imóveis não aparecem no mapa?
- Como o código verifica se existe coordenada?

Grupo 2 — Mapa de Valor

Criar uma tela que mostre imóveis por faixa de valor.

Entrega mínima

- Função para classificar valor
- Cores diferentes por faixa
- Legenda
- Popup com valor
- Filtro por bairro ou logradouro

Perguntas que o grupo deve saber responder

- Como definimos as faixas?
- O que acontece se o valor estiver vazio?
- Como o filtro muda o mapa?

Grupo 3 — Dashboard

Criar gráficos e indicadores.

Entrega mínima

- Total de registros
- Valor médio por m²
- Área total dos terrenos
- Gráfico por bairro
- Ranking de logradouros

Perguntas que o grupo deve saber responder

- Como calcular média?
- Como agrupar por bairro?
- Qual gráfico comunica melhor a informação?

Grupo 4 — Qualidade da Base

Identificar problemas nos dados.

Entrega mínima

- Total sem coordenadas
- Total sem CEP
- Total sem valor
- Tabela de inconsistências
- Score de qualidade

Perguntas que o grupo deve saber responder

- Por que qualidade de dados importa?
- O que é um campo obrigatório?
- Como calcular um percentual de preenchimento?

13. Uso orientado de Inteligência Artificial

A IA será permitida como ferramenta de apoio, mas não como substituta do aprendizado. Os alunos poderão pedir explicações, exemplos, correções e sugestões. Porém, todo código entregue deverá ser compreendido pelo grupo. Na apresentação, qualquer aluno poderá ser questionado sobre uma função usada no projeto.

13.1 Uso permitido

- Entender erro de código.
- Pedir explicação linha por linha.
- Gerar exemplo simples.
- Melhorar visual.
- Revisar função.
- Criar comentários no código.
- Sugerir nomes de variáveis.
- Comparar soluções.

13.2 Uso não permitido

- Entregar código que o grupo não entende.

- Pedir para a IA fazer todo o trabalho sem adaptação.
- Copiar resposta sem testar.
- Esconder que usou IA.
- Usar IA para inventar dados.
- Apagar partes do projeto sem entender.

14. Prompts prontos para usar com IA

14.1 Para entender um erro

Estou desenvolvendo uma atividade de Computação II usando HTML, CSS e JavaScript puro.

Meu objetivo é trabalhar com uma lista de imóveis em JSON.

Estou recebendo este erro:

[cole aqui o erro]

Este é o trecho do meu código:

[cole aqui o código]

Explique o erro de forma simples, como para um aluno iniciante, e sugira uma correção sem reescrever o projeto inteiro.

14.2 Para entender uma função

Explique este código linha por linha, como se eu estivesse aprendendo JavaScript agora.

Código:

[cole aqui o código]

Depois, diga:

1. O que a função recebe;
2. O que ela faz;
3. O que ela retorna;
4. Onde posso usar essa função no meu projeto de dashboard imobiliário.

14.3 Para criar uma função simples

Estou criando um painel imobiliário com JavaScript puro.

Tenho um array de objetos chamado dados. Cada objeto possui os campos:

```
bairro
valor_m2_testada
area_terreno
valor_patrimonial
latitude
longitude
```

Crie uma função simples para [descrever a tarefa].

Requisitos:

- usar JavaScript puro;
- código simples para aluno iniciante;
- comentar as principais linhas;
- não usar framework;
- explicar depois como testar a função.

14.4 Para adaptar código ao projeto

Tenho este código:

[cole aqui o código]

Quero adaptar para o meu projeto, onde os dados dos imóveis estão no array `PGV_DATA` e os campos são:

[cole aqui os nomes dos campos]

Adapte o código mantendo JavaScript puro e explique o que foi alterado.

14.5 Para criar gráfico

Estou usando Chart.js em um projeto com HTML, CSS e JavaScript puro.

Tenho um array de imóveis chamado `dados`, com os campos `bairro` e `valor_m2_testada`.

Quero criar um gráfico de barras com o valor médio do m^2 por bairro.

Crie um exemplo simples, separado em:

1. Função para agrupar os dados;
2. Função para calcular a média;
3. Código do gráfico Chart.js;
4. HTML necessário.

Explique como integrar ao meu projeto.

14.6 Para criar mapa

Estou usando Leaflet em um projeto com HTML, CSS e JavaScript puro.

Tenho um array de imóveis com `latitude`, `longitude`, `bairro`, `endereço` e `valor_m2_testada`.

Crie um exemplo simples para:

1. Inicializar o mapa em Matinhos/PR;
2. Percorrer os dados;
3. Verificar se latitude e longitude existem;
4. Criar marcadores;
5. Criar popup com bairro, endereço e valor.

Use código simples e explique linha por linha.

14.7 Para revisar antes de entregar

Revise este código como professor de Computação II.

Verifique:

1. Se há erro de sintaxe;
2. Se as variáveis têm nomes claros;
3. Se o código está simples para iniciantes;
4. Se há repetição desnecessária;
5. Se há risco de quebrar quando algum dado estiver vazio.

Código:

[cole aqui o código]

Sugira melhorias sem complicar demais.

15. Critérios de avaliação

Critério	Pontuação	O que será observado
Funcionamento técnico	30 pontos	Carrega dados, exhibe informações, não quebra com dados ausentes, apresenta resultado coerente.
Organização do código	20 pontos	Nomes claros, funções pequenas, comentários úteis, separação entre HTML, CSS e JS.
Interpretação dos dados	20 pontos	O grupo entende os campos usados, sabe explicar indicadores e reconhece limitações da base.
Uso responsável da IA	15 pontos	Registra prompts, explica o que aproveitou, adapta o código e demonstra compreensão.
Apresentação extensionista	15 pontos	Explica o problema, mostra utilidade para imobiliárias, demonstra a ferramenta e responde perguntas.

16. Níveis de entrega

Nível básico — Obrigatório

- Carregar dados.
- Exibir informação na tela.
- Usar pelo menos uma função.
- Mostrar resultado funcional.
- Explicar o código.

Nível intermediário — Desejável

- Criar gráfico.

- Criar tabela.
- Criar filtro.
- Calcular média.
- Agrupar por bairro.

Nível avançado — Extra

- Mapa com cores.
- Filtros combinados.
- Score de qualidade.
- Ranking dinâmico.
- Layout mais profissional.

17. Entregas finais por grupo

- Código do módulo desenvolvido.
- Pequena documentação.
- Registro do uso de IA.
- Apresentação de 5 minutos.
- Lista de melhorias futuras.

Modelo de documentação do grupo

```
Nome do grupo:  
Integrantes:  
Módulo desenvolvido:  
Objetivo do módulo:  
Campos utilizados:  
Funções criadas:  
Dificuldades encontradas:  
Como a IA ajudou:  
Melhorias futuras:
```

18. Conexão com imobiliárias e comunidade externa

Para fortalecer o caráter extensionista, os grupos poderão conversar com imobiliárias, corretores ou atores locais para entender que informações seriam úteis em um painel imobiliário. Essa conversa pode ser presencial, por telefone, mensagem ou em uma apresentação final.

Roteiro de entrevista com imobiliárias

1. Quais informações vocês mais usam para avaliar uma região?
2. Bairro e rua influenciam muito no valor?
3. Um mapa de valores ajudaria na análise comercial?
4. Quais filtros seriam úteis?
5. Vocês usam alguma ferramenta parecida hoje?
6. Que tipo de gráfico ou indicador seria interessante?
7. A qualidade dos dados é um problema no mercado imobiliário?
8. Que melhoria vocês sugeririam para a ferramenta?

19. Roteiro da apresentação final

Cada grupo deverá apresentar em aproximadamente 5 minutos:

- Qual parte do sistema desenvolveu.
- Que problema o módulo resolve.
- Quais dados foram usados.
- Como funciona a tela ou funcionalidade.
- Quais dificuldades foram encontradas.
- Como a IA ajudou no desenvolvimento.
- Que melhoria futura seria importante.

20. Resultado esperado de aprendizagem

Ao final da atividade, espera-se que o aluno consiga afirmar que participou do desenvolvimento de uma aplicação web real, usando dados imobiliários, JavaScript, JSON, mapas, gráficos e análise de qualidade dos dados. Também deverá demonstrar que aprendeu a usar IA como apoio para entender erros, revisar código e melhorar a solução, sem deixar de compreender o que foi entregue.

Mensagem final aos alunos

Não é necessário saber tudo no início. O mais importante é aprender a dividir o problema, testar pequenas partes, pedir ajuda de forma correta, documentar o processo e conseguir explicar o que foi construído.

21. Prazo de entrega

Prévia do trabalho: 19/mai

Entrega: **09/jun – 18:30** (presença de corretores locais), via *teams*.

22. Sugestão

As próximas 4 imagens são sugestões de interfaces.

SPU PGV Matinhos
Imoveis da Uniao

- Mapa Geral
- Mapa de Valor
- Dashboard
- Qualidade da Base

PLANTA GENERICA DE VALORES

Planta Genérica de Valores – Matinhos/PR

4911
registros

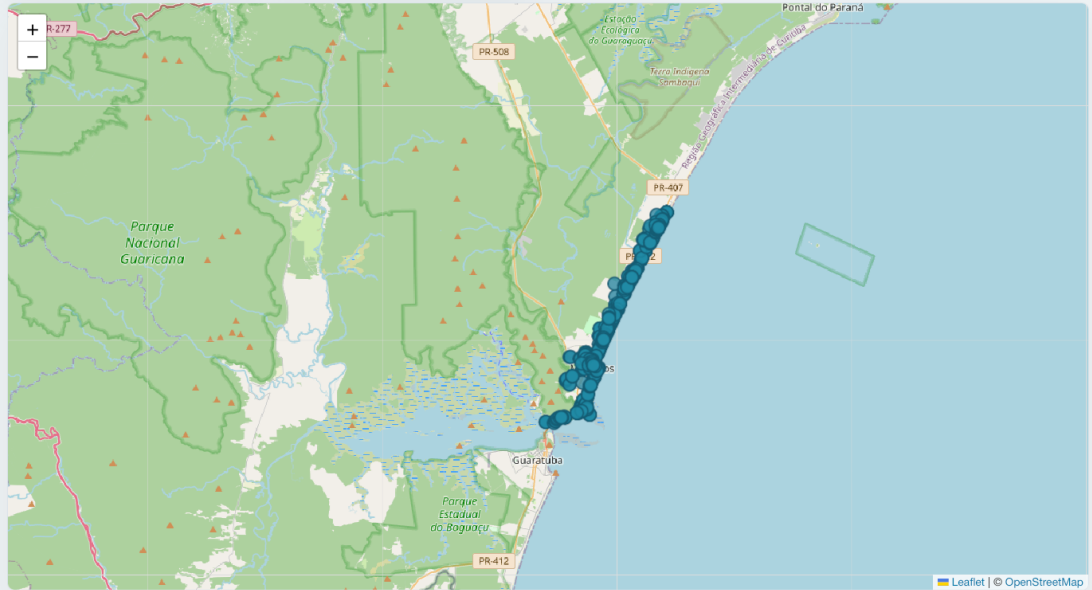
Painel geoespacial e analítico para apoio à avaliação patrimonial

Pontos no mapa

4.718

Sem coordenadas

193



SPU PGV Matinhos
Imoveis da Uniao

- Mapa Geral
- Mapa de Valor
- Dashboard
- Qualidade da Base

PLANTA GENERICA DE VALORES

Planta Genérica de Valores – Matinhos/PR

4911
registros

Painel geoespacial e analítico para apoio à avaliação patrimonial

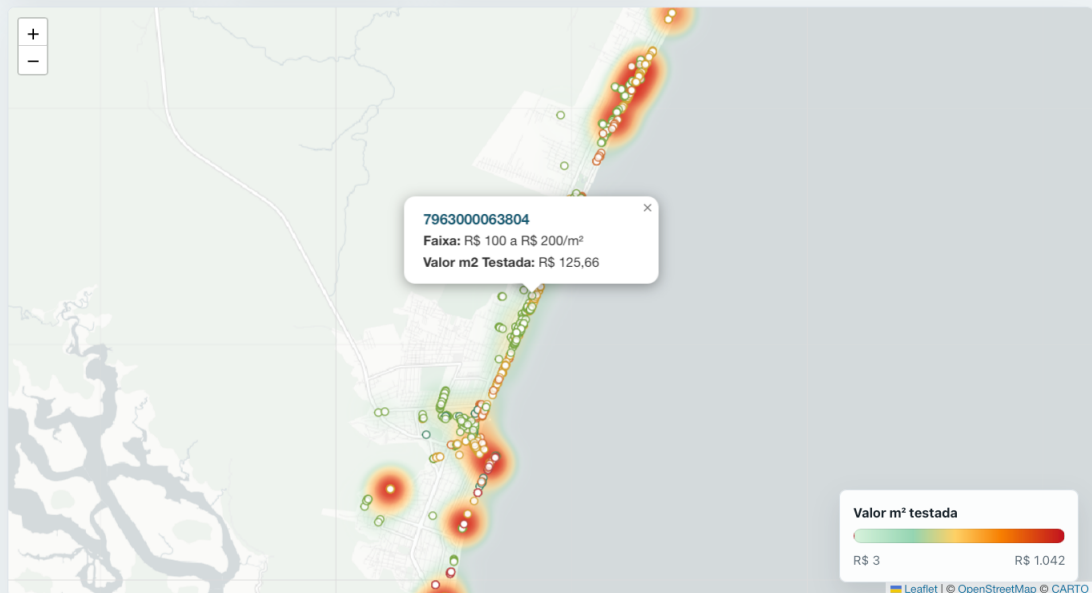
Pontos filtrados

4.718

Pontos de consulta

Abrangência

26px



SPU PGV Matinhos
Imoveis da Uniao

- Mapa Geral
- Mapa de Valor
- Dashboard**
- Qualidade da Base

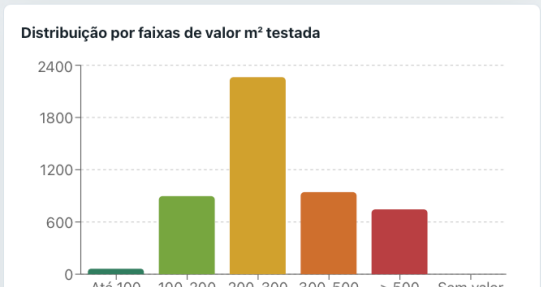
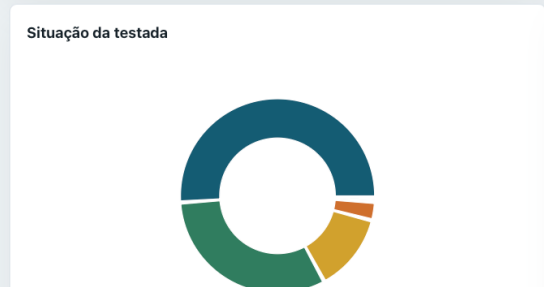
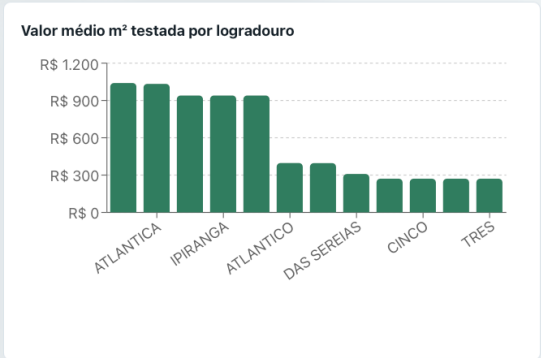
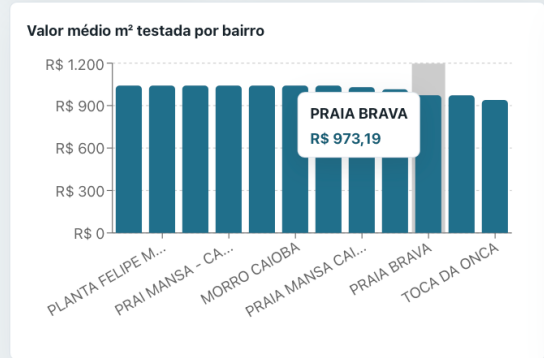
PLANTA GENERICA DE VALORES

Planta Genérica de Valores – Matinhos/PR

4911
registros

Painel geoespacial e analítico para apoio à avaliação patrimonial

Total de registros 4.911	Total de imóveis 4.911	Total de utilizações 4.911	Área total dos terrenos 41.748.484,7 m²	Área total construída 616.923,31 m²
Valor patrimonial total R\$ 238,66 mi R\$ 238.659.950,33	Valor médio m ² testada R\$ 359,96	Com coordenadas 4.718	Sem coordenadas 193	



SPU PGV Matinhos
Imoveis da Uniao

- Mapa Geral
- Mapa de Valor
- Dashboard
- Qualidade da Base**

PLANTA GENERICA DE VALORES

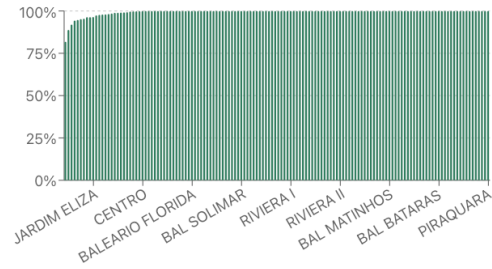
Planta Genérica de Valores — Matinhos/PR

4911
registros

Painel geoespacial e analítico para apoio à avaliação patrimonial

Score geral da base 99,3%	Inconsistências mapeadas 386			
Sem latitude/longitude 193	Sem GeoJSON 193	Sem CEP 0	Sem bairro 0	Sem endereço 0
Sem valor m² testada 0	Sem valor patrimonial 0	Sem área do terreno 0	Sem data de avaliação 0	

Score médio por bairro



Registros de menor qualidade

RIP IMÓVEL	BAIRRO	ENDEREÇO
7963010383160	BAL RIVIERA	AV BEIRA MAR 2277 APTO 404
7963010383089	BAL RIVIERA	AV BEIRA MAR 2277 APTO 403
7963010382945	BAL RIVIERA	AV BEIRA MAR 2277 APTO 402
7963010382864	BAL RIVIERA	AV BEIRA MAR 2277 APTO 401
7963010382783	BAL RIVIERA	AV BEIRA MAR 2277 APTO 304
7963010382600	BAL RIVIERA	AV BEIRA MAR 2277 APTO 303
7963010382511	BAL RIVIERA	AV BEIRA MAR 2277 APTO 302
7963010382430	BAL RIVIERA	AV BEIRA MAR 2277 APTO 301
7963010382350	BAL RIVIERA	AV BEIRA MAR 2277 APTO 204
7963010382279	BAL RIVIERA	AV BEIRA MAR 2277 APTO 203
7963010382198	BAL RIVIERA	AV BEIRA MAR 2277 APTO 202