

# WaterWise - Plano de Entregas Global Solution 2025

*Organização completa das entregas por disciplina*



## CRONOGRAMA DE ENTREGAS

*Deadline: 06/06/2025 até 23h55*

### Semana 1-2 (26/05 - 02/06)

- ☒ **IoT System Development** (Disruptive Architectures)
- ☒ **Database Design & PL/SQL** (Mastering Database)
- ☒ **API .NET Core** (Advanced Business Development)

### Semana 3 (03/06 - 04/06)

- ☒ **Java Spring Application** (Java Advanced)
- ☒ **React Native Mobile App** (Mobile Development)

### Semana 4 (05/06 - 06/06)

- ☒ **Azure Cloud Deploy** (DevOps Tools)
- ☒ **SCRUM Planning & Tests** (Compliance & QA)
- ☒ **Final Documentation & Videos**



## ESTRUTURA DE ARQUIVOS PARA ENTREGA

**Template de arquivo .txt para cada disciplina:**

INTEGRANTES:

- [Nome Completo 1] - RM: [12345] - Turma: [2TDSB]
- [Nome Completo 2] - RM: [12346] - Turma: [2TDSB]
- [Nome Completo 3] - RM: [12347] - Turma: [2TDSB]

SOLUÇÃO: WaterWise - Sistema Inteligente de Prevenção a Enchentes Urbanas

DESCRIÇÃO: Sistema IoT integrado que monitora propriedades rurais para prevenir enchentes urbanas através da otimização da retenção hídrica do solo, utilizando sensores, IA e análise preditiva em tempo real.

REPOSITÓRIOS GITHUB:

- API .NET: <https://github.com/waterwise-team/waterwise-api-dotnet>
- Mobile App: <https://github.com/waterwise-team/waterwise-mobile-react>
- Java Admin: <https://github.com/waterwise-team/waterwise-admin-spring>
- IoT System: <https://github.com/waterwise-team/waterwise-iot-sensors>

VÍDEOS YOUTUBE:

- Pitch (3min): <https://youtu.be/waterwise-pitch-2025>
- Demo Completa: <https://youtu.be/waterwise-demo-complete>
- IoT Demo: <https://youtu.be/waterwise-iot-demo>

DEPLOYS PRODUÇÃO:

- API: <https://waterwise-api.azurewebsites.net>
- Admin: <https://waterwise-admin.azurewebsites.net>



## ENTREGAS POR DISCIPLINA



### 1 ADVANCED BUSINESS DEVELOPMENT WITH .NET

**Entrega:** Arquivo ZIP contendo:

```
waterwise-dotnet-api.zip
├── WaterWise.API/
│   ├── Controllers/
│   ├── Models/
│   ├── Services/
│   ├── Data/
│   └── Program.cs
├── WaterWise.Tests/
│   └── Controllers.Tests/
├── README.md (documentação completa)
├── waterwise-api.sln
└── integrantes.txt
```

### README.md deve conter:

- Descrição do Projeto WaterWise
  - Tecnologias utilizadas (.NET 8, EF Core, ML.NET, RabbitMQ)
  - Como Executar o Projeto (passo a passo)
  - Documentação dos Endpoints (Swagger)
  - Instruções de testes (xUnit)
  - Arquitetura de microsserviços
  - Exemplos de uso da API
- 

## 2 COMPLIANCE, QUALITY ASSURANCE & TESTS

**Entrega:** Arquivo PDF contendo:

```
WaterWise_SCRUM_Tests.pdf
├── 1. PITCH DO PROJETO (1 página)
├── 2. AZURE BOARDS - BACKLOG SCRUM
│   ├── Screenshot do backlog organizado
│   ├── Épicas, Features e PBIs detalhados
│   ├── Critérios de aceite de cada item
│   └── Priorização por sprints
├── 3. PLANO DE TESTES MANUAIS
│   ├── Lista de 10+ testes planejados
│   ├── Dados de entrada/saída para cada teste
│   ├── Procedimentos detalhados (passo a passo)
│   └── Status de execução com responsáveis
├── 4. LINK DO PROJETO AZURE
│   └── https://dev.azure.com/waterwise/WaterWise-2025
```

**Exemplo de Teste Documentado:**

### TESTE 003 - Alerta Automático Enchente

Objetivo: Verificar se sistema gera alerta quando risco de enchente é detectado

Dados Entrada:

- Umidade solo: 15% (crítica)
- Previsão chuva: 80mm nas próximas 2h
- Área propriedade: 50 hectares

Dados Saída Esperada:

- Alerta criado com severidade HIGH
- Notificação enviada ao proprietário
- Dashboard atualizado em tempo real

Procedimento:

1. Simular leituras de sensores via MQTT
2. Aguardar processamento do algoritmo ML (máx 30s)
3. Verificar criação do alerta na tabela 'alertas'
4. Confirmar recebimento da notificação
5. Validar exibição no dashboard web

Executado por: Maria Santos - 25/05/2025

Status: PASSOU 

Observações: Alerta gerado em 12s, notificação recebida corretamente

---

## 3 DEVOPS TOOLS & CLOUD COMPUTING

**Entrega:** Arquivo PDF + Vídeo demonstração

**PDF deve conter:**

WaterWise\_Azure\_Deploy.pdf

- CAPA (Nome do grupo, integrantes, RMs)
- 1. DESCRIÇÃO DO PROJETO (resumo executivo)
- 2. ARQUITETURA DEVOPS (diagrama Draw.io)
  - Frontend (React/Next.js)
  - Backend (.NET API)
  - Database (Azure SQL)
  - IoT Hub (sensor data)
  - Application Insights (monitoring)
- 3. RECURSOS AZURE CRIADOS
  - App Service Plan
  - Web Apps (API + Frontend)
  - SQL Database
  - Storage Account
  - Application Insights
- 4. LINK GITHUB + README
- 5. JSON APIs (GET/POST/PUT/DELETE examples)

## Vídeo Demonstração (5-10min):

1. **[0-2min]** Tour pelos recursos no Azure Portal
  2. **[2-4min]** Deploy automático via GitHub Actions
  3. **[4-6min]** Teste das APIs em produção (Postman/Swagger)
  4. **[6-8min]** Demonstração CRUD completo
  5. **[8-10min]** Monitoramento e logs em Application Insights
- 

## 4 DISRUPTIVE ARCHITECTURES: IOT

**Entrega:** Arquivo TXT + Repositório GitHub + Vídeo YouTube

### Arquivo TXT:

WaterWise\_IoT\_Links.txt

INTEGRANTES:

[Lista completa com RMs e turmas]

REPOSITÓRIO GITHUB:

<https://github.com/waterwise-team/waterwise-iot-sensors>

VÍDEO YOUTUBE (3min):

<https://youtu.be/waterwise-iot-demo-2025>

DESCRIÇÃO:

Sistema IoT com ESP32, 4 sensores (umidade solo, temperatura, pluviômetro, atuador bomba), comunicação MQTT, dashboard Node-RED e integração ThingSpeak para monitoramento em tempo real de propriedades rurais.

SIMULAÇÃO WOKWI:

<https://wokwi.com/projects/waterwise-sensors-demo>

DASHBOARD NODE-RED:

<http://waterwise-nodered.herokuapp.com/ui>

### Repositório GitHub deve ter:

- Código completo Arduino/ESP32
- Esquemas de ligação (Fritzing/Tinkercad)
- Fluxos Node-RED exportados
- README detalhado com instruções
- Imagens do protótipo funcionando

- Configurações MQTT e ThingSpeak
- 

## 5 JAVA ADVANCED

**Entrega:** Documento PDF + Links + Vídeos

**Documento deve conter:**

WaterWise\_Java\_Spring.pdf

- 1. LINKS DOS REPOSITÓRIOS
- 2. LINKS DOS DEPLOYS EM PRODUÇÃO
- 3. INSTRUÇÕES DE ACESSO E TESTES
  - Usuários de demonstração
  - Funcionalidades implementadas
  - Como testar OAuth2
- 4. VÍDEO DEMONSTRAÇÃO (10min)
- 5. VÍDEO PITCH (3min)

**Funcionalidades a demonstrar:**

- Login OAuth2 (Google/Microsoft)
  - Dashboard com dados em tempo real
  - CRUD de propriedades com validação
  - Sistema de alertas automáticos
  - Relatórios gerados por Spring AI
  - Internacionalização (PT/EN)
  - Integração RabbitMQ
  - Testes unitários funcionando
- 

## 6 MASTERING DATABASE

**Entrega:** PDF + Vídeo explicativo

**PDF deve conter:**

## WaterWise\_Database.pdf

- 1. MODELO RELACIONAL (DER + DDL)
- 2. PACKAGE COMPLETA PKG\_WATERWISE
  - Código SQL completo
  - Screenshots de execução
  - Resultados dos testes
- 3. CONSULTAS SQL AVANÇADAS
  - Relatórios com JOINS complexos
  - Funções de agregação
  - CTEs e Window Functions
- 4. MONGODB INTEGRATION
  - Estrutura dos documentos
  - Queries de exemplo
  - Screenshots da integração
- 5. EXEMPLOS DE EXECUÇÃO

### Vídeo deve mostrar:

- Execução das procedures e functions
- Triggers em funcionamento
- Consultas SQL complexas rodando
- Integração com MongoDB
- Dados sendo inseridos via aplicação

---

## **7 MOBILE APP DEVELOPMENT**

**Entrega:** Arquivo TXT + Repositório GitHub + Vídeo

**Arquivo TXT:**

WaterWise\_Mobile\_App.txt

REPOSITÓRIO GITHUB:

<https://github.com/waterwise-team/waterwise-mobile-react>

INTEGRANTES E DESCRIÇÃO:

[Conforme template padrão]

TECNOLOGIAS:

- React Native + Expo
- Firebase Authentication
- Axios para consumo de APIs
- React Navigation
- Design personalizado sustentável

FUNCIONALIDADES:

- 5+ telas com navegação fluida
- Login/cadastro Firebase
- CRUD completo integrado com API .NET/Java
- Dashboard com dados em tempo real
- Mapa interativo das propriedades
- Sistema de notificações push
- Interface responsiva e acessível

APK DEMO (se possível):

<https://expo.dev/@waterwise/waterwise-mobile>

### **Vídeo Demonstração (5min):**

1. **[0-1min]** Login e navegação entre telas
2. **[1-2min]** CRUD de propriedades
3. **[2-3min]** Dashboard e visualização de dados
4. **[3-4min]** Sistema de alertas e notificações
5. **[4-5min]** Integração com APIs e funcionamento offline

---

## **LINKS DE CADASTRO E ENTREGA**

### **Cadastro de Grupos (até 26/05):**

<https://forms.gle/XphT9RV1aAFT6swL8>

### **Portal de Entrega FIAP:**







1. Acessar: [www2.fiap.com.br](http://www2.fiap.com.br)



2. Login → Aulas → Entrega de Trabalhos
  3. Localizar trabalhos "Global Solution"
  4. Upload dos arquivos por disciplina
  5. Confirmar entrega até 06/06 23h55
- 

## CRITÉRIOS DE AVALIAÇÃO

### Para ganhar a premiação:

-  **Notas 9+ em todas as disciplinas**
-  **Vídeo pitch Java Advanced nota 9+**
-  **Solução integrada e funcional**
-  **Impacto real no problema proposto**
-  **Qualidade técnica excepcional**
-  **Documentação completa e profissional**

### Diferencial WaterWise:

1. **Problema Real:** Mairiporã tem 26 áreas de risco documentadas
  2. **Solução Inovadora:** Prevenção rural → proteção urbana
  3. **Tech Stack Completa:** IoT + IA + Cloud + Mobile integrados
  4. **Impacto Mensurável:** Solo saudável = 20x mais absorção
  5. **Escalabilidade:** Replicável para qualquer região metropolitana
- 


## CHECKLIST FINAL DE ENTREGA

### Antes da entrega, verificar:

- ☐ Todos os repositórios GitHub públicos e documentados
- ☐ READMEs completos em todos os projetos
- ☐ Vídeos postados no YouTube (público/não listado)
- ☐ Deploys funcionando em produção
- ☐ Arquivo .txt em cada entrega ZIP
- ☐ Links testados e funcionais
- ☐ Documentação em PDF bem formatada
- ☐ Código limpo e comentado
- ☐ Testes unitários passando
- ☐ Integração entre sistemas funcionando

### No dia da entrega (06/06):

- ☐ Upload de todos os arquivos antes das 23h55
  - ☐ Confirmação de entrega em cada disciplina
  - ☐ Backup de todos os arquivos localmente
  - ☐ Screenshots de confirmação das entregas
  - ☐ Comunicação com equipe sobre status
- 

 **WaterWise representa a convergência perfeita entre necessidade real e inovação tecnológica. Não é apenas um projeto acadêmico - é uma solução que pode salvar vidas e proteger comunidades inteiras dos impactos crescentes das mudanças climáticas.**

 **Cada gota conta. Cada propriedade importa. Cada algoritmo faz diferença.**

 **WaterWise: Transformando dados em proteção, tecnologia em esperança.**