



TREINAMENTOS, CONSULTORIA
& DESENVOLVIMENTO

2020



ESPECIALISTAS EM C++ E Qt

<http://qmob.solutions>

Por que C++?

A linguagem C++ (<http://isocpp.org>) é atualmente utilizada por milhões de desenvolvedores em todo o mundo e a decisão pela utilização dessa linguagem em muitos projetos se dá por diversos fatores: disponibilização de abstrações de alto nível com excelente desempenho, acesso a baixo nível quando necessário, aplicabilidade em diversos domínios, alta portabilidade, *overhead* zero (sem penalidades adicionais ao não utilizar recursos avançados da linguagem), excelente gerenciamento de recursos e dominância na indústria. Com a modernização da linguagem de programação nas normas C++11/14/17/20, sua adoção é consideravelmente ampliada, principalmente nas áreas de *mobile* e *cloud computing*.

Por que Qt?

O Qt (<http://www.qt.io>) é um *toolkit* para desenvolvimento multiplataforma de software adotado por mais de 70 indústrias em todo o mundo, nas áreas de *desktop*, *mobile*, sistemas embarcados e IoT (*Internet of Things*). O Qt permite o desenvolvimento de sistemas para diversas plataformas tais como Windows, GNU/Linux, OS X, Android e iOS com um único código-fonte, apresenta vasta documentação, alta produtividade e excelente desempenho mesmo em plataformas mais modestas.



SOBRE A QMOB SOLUTIONS

A **qmob solutions** (<http://qmob.solutions>) é uma empresa especializada em treinamentos, consultoria e desenvolvimento de soluções baseadas nas tecnologias C++ e Qt. Criada por especialistas C++ e Qt com 20 anos de experiência, a **qmob solutions** preza pela oferta de treinamentos adaptados às necessidades do cliente e pelo desenvolvimento de soluções corretas e eficazes sob o ponto de vista da Engenharia de Software, reduzindo custos de manutenção e atendendo os requisitos não-funcionais mais importantes do projeto. Nosso foco é a entrega de soluções elegantes e funcionais para múltiplas plataformas: Windows, Linux, OS X, iOS, Android e plataformas embarcadas, agregando os diversos benefícios que o C++ e o Qt possuem ao manter um único código-fonte para múltiplas plataformas. A **qmob solutions** é também especialista na definição e implantação de processos maduros de desenvolvimento, de QA (*Quality Assurance*) e de CI/CD (*Continuous Integration/Continuous Delivery*), unindo agilidade e qualidade de processo e produto.

Conforme apresentado em <https://www.qt.io/contact-us/partners>, a **qmob solutions** é o primeiro e único *Service Partner* oficial da The Qt Company (empresa que atualmente desenvolve o Qt) no Brasil e na América-Latina. Com isso, a **qmob solutions** estreita ainda mais o seu relacionamento com a comunidade global do Qt, garantindo treinamentos sempre atualizados, consultorias eficazes e produtivas, bem como acesso às novidades mais recentes do Qt.



SUMÁRIO

Treinamentos

Fundamentos de C++	6
Desenvolvendo Aplicações Gráficas com QtWidgets	8
Fundamentos de QML	10
Aplicações Cliente-Servidor com Qt e RESTful	12
Desenvolvendo Aplicações Android com Qt	14
Padrões de Projeto com Qt	16
Computação Gráfica com Qt 3D	18
Sistemas Embarcados com Qt	20
Otimizando Aplicações Qt	22
C++ Avançado	24
QML Avançado	26
Gerência de Qualidade, DevOps e Continuous Delivery para Projetos Qt	28
Qt for Python	30
Desenvolvimento de Jogos com Qt	32



Consultoria

Projeto e Análise Arquitetural	34
Processos de Desenvolvimento e Gerência de Qualidade	35
Planejamento de Migração para Qt	36
Análise de Licenciamento Qt	37

Desenvolvimento

Desenvolvimento Outsource de Software	38
---------------------------------------	----



FUNDAMENTOS DE C++

Visão Geral e Objetivos

A linguagem C++ é uma poderosa ferramenta para a construção de sistemas flexíveis, integráveis e com excelente desempenho. Seus mecanismos de abstração e alto grau de portabilidade fazem do C++ a linguagem de escolha para desenvolvimento de sistemas em áreas tais como manufatura, alto desempenho, *cloud computing* e aplicativos móveis.

Neste treinamento, são apresentados os fundamentos básicos do C++ e como os principais recursos da Orientação a Objetos são nela utilizados. Ao final do treinamento, o aluno deverá ser capaz de aplicar, de forma correta e sistemática, os principais recursos da linguagem; construindo sistemas de fácil manutenção, alto desempenho e passível de execução em múltiplas plataformas.



ESPECIALISTAS EM C++ E Qt

<http://qmob.solutions>

Pré-requisitos

- ◆ Lógica de programação
- ◆ Noções de Orientação a Objetos (desejável)

Carga Horária

40h

Método

- ◆ Fundamentação expositiva
- ◆ Práticas em laboratório

Conteúdo

- ◆ Visão geral da linguagem
- ◆ Objetos e classes
- ◆ Funções-membro e dados-membro
- ◆ Operadores de visibilidade e encapsulamento
- ◆ Construtores, destrutores e *copy constructor*
- ◆ Agregação e composição
- ◆ Herança de implementação
- ◆ Sobreposição de funções-membro
- ◆ Herança de interfaces (*sub-typing*)
- ◆ Funções virtuais, polimorfismo e ligação dinâmica
- ◆ Interfaces e classes abstratas

DESENVOLVENDO APLICAÇÕES GRÁFICAS COM QtWIDGETS

Visão Geral e Objetivos

O Qt é um *toolkit* para desenvolvimento multiplataforma de sistemas, criado há mais de 23 anos e amplamente utilizado em indústrias de todo o mundo. O QtWidgets é o módulo do Qt para criação de interfaces gráficas de usuário utilizando a linguagem C++ e oferece um amplo conjunto de funcionalidades para criação de workflows complexos e interfaces sofisticadas.

Neste treinamento, são apresentados os conceitos básicos do Qt (utilizados não somente em aplicações visuais) e seus principais recursos para desenvolvimento multiplataforma de interfaces gráficas de usuário. Ao final do treinamento, o aluno deverá ser capaz de implementar, de forma produtiva, sistemas visuais passíveis de serem executados em diferentes plataformas.

Pré-requisitos

- ◆ Fundamentos de C++
- ◆ Noções de programação de sistemas visuais (desejável)

Carga Horária

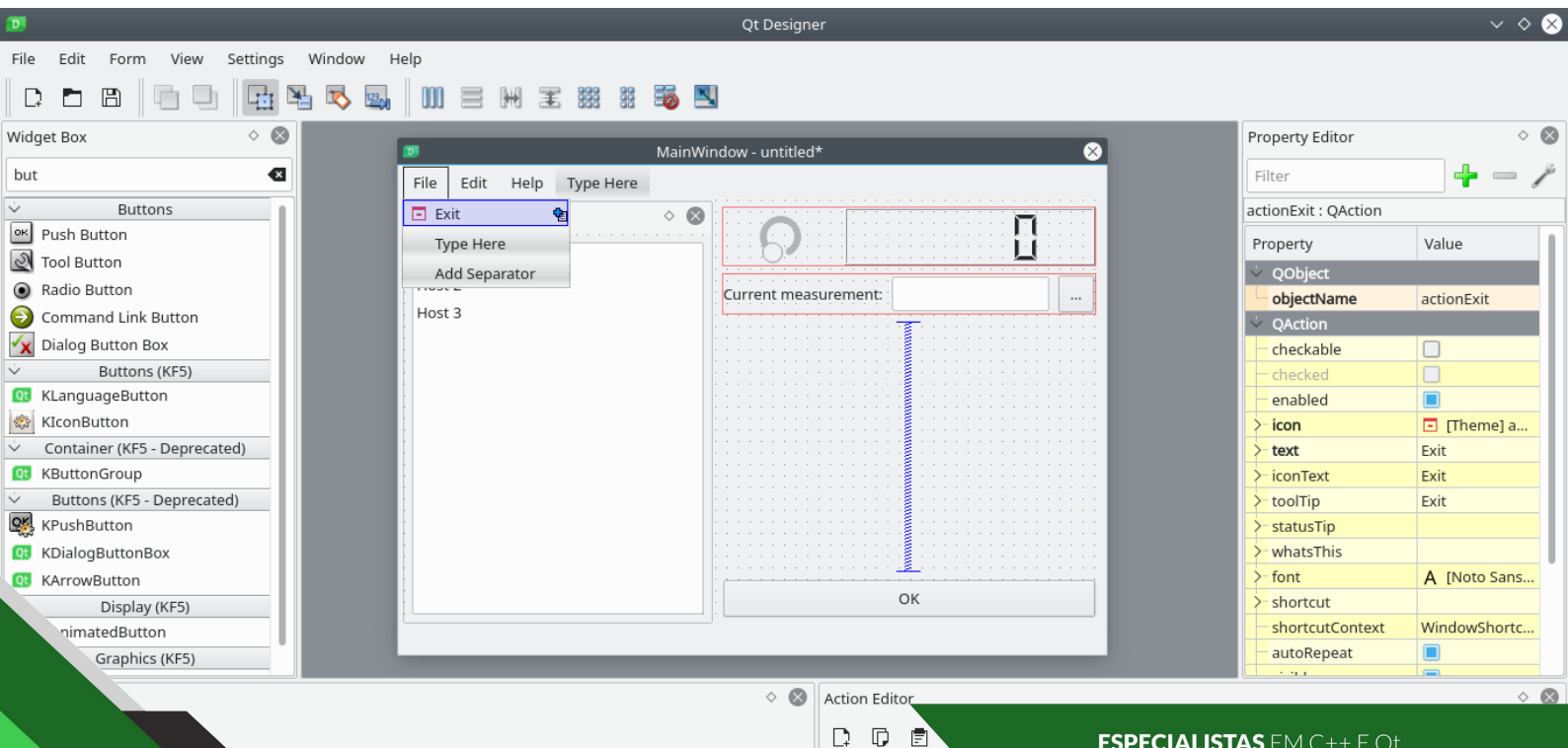
40h

Método

- ◆ Fundamentação expositiva
- ◆ Práticas em laboratório

Conteúdo

- ◆ Visão geral do Qt
- ◆ MOC (*Meta-Object Compiler*) e o modelo de objetos do Qt
- ◆ Sinais e *slots*
- ◆ Propriedades (dinâmicas)
- ◆ Metaobjetos, parentesco e *memory ownership*
- ◆ UIC (*User Interface Compiler*) e o Qt Designer
- ◆ Janelas principais, *layouts* e *dialogs*
- ◆ Programação com *model/view*
- ◆ *Containers* do Qt
- ◆ Acesso a banco de dados
- ◆ Entrada/Saída com XML e JSON



FUNDAMENTOS DE QML

Visão Geral e Objetivos

O QML é uma linguagem declarativa para criação de interfaces gráficas e é parte integrante do toolkit Qt. Dentre suas principais vantagens, destacam-se: baixa curva de aprendizagem, alta produtividade, expressividade na construção de interfaces gráficas para *tablets* e *smartphones*, alto desempenho em função do suporte à execução em GPUs e facilidade de integração com linguagens tais como JavaScript e C++.

Neste treinamento, são apresentados os fundamentos da linguagem QML e seus principais recursos para criação de interfaces modernas. Ao final do treinamento, o aluno deverá ser capaz de projetar e implementar, de forma produtiva, interfaces gráficas adequadas à execução em múltiplos dispositivos.

Pré-requisitos

- ◆ Fundamentos de C++
- ◆ Noções de programação de sistemas visuais (desejável)

Carga Horária

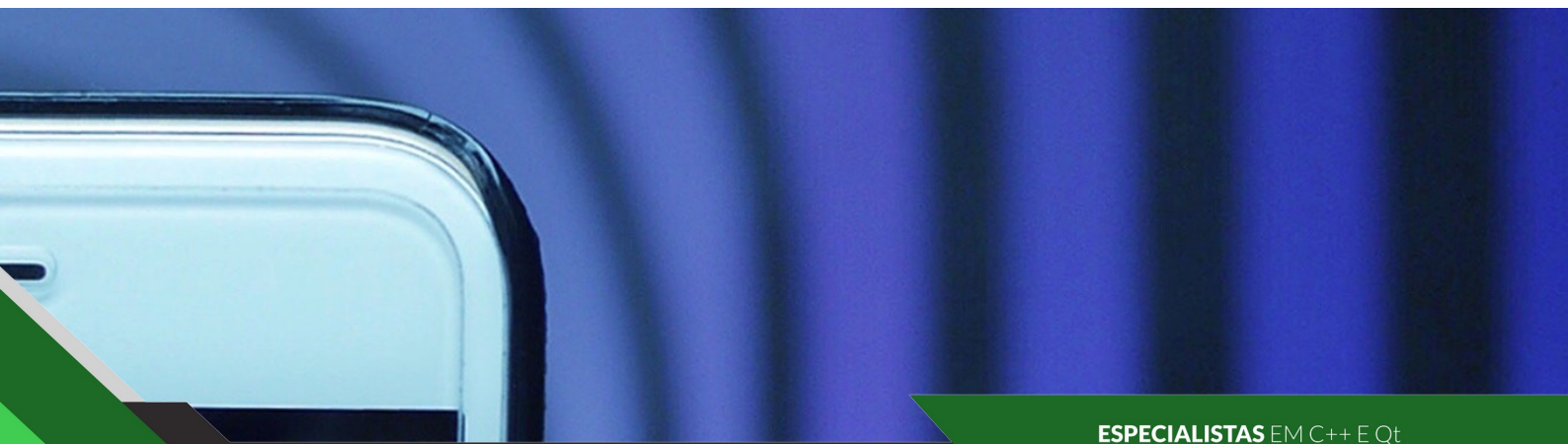
40h

Método

- ◆ Fundamentação expositiva
- ◆ Práticas em laboratório

Conteúdo

- ◆ Introdução ao Qt, QML e QtQuick
- ◆ Sintaxe básica da QML (sentenças *import*, declaração de objetos, comentários)
- ◆ Atributos de objetos QML
- ◆ *Property binding*
- ◆ Integrando QML e JavaScript
- ◆ Sistema de Tipos do QML
- ◆ Documentos e módulos
- ◆ Lidando com entrada de dados
- ◆ O QtQuickControls 2
- ◆ Posicionando elementos (*bindings*, manual, âncoras, *positioners* e *layouts*)
- ◆ *Model/View* com QML
- ◆ Componentes QML e instanciação dinâmica
- ◆ Animações e máquinas de estado
- ◆ Internacionalização e localização



APLICAÇÕES CLIENTE-SERVIDOR COM Qt e RESTful

Visão Geral e Objetivos

O RESTful é uma tecnologia amplamente utilizada na implementação de serviços web, tem como base o protocolo HTTP e é influenciada pelo estilo arquitetural REST. Projetar aplicações cliente-servidor com funcionalidades de *back-end* implementadas como serviços RESTful tem sido um padrão arquitetural comum, em domínios tais como aplicativos móveis, sistemas web e aplicações para *cloud computing*.

Neste treinamento, são apresentados os recursos do Qt para invocação de serviços RESTful e manipulação de dados XML e JSON, bem como técnicas para implementação de serviços RESTful. Ao final do treinamento, o aluno deverá ser capaz de projetar e implementar sistemas cliente-servidor utilizando o Qt.



ESPECIALISTAS EM C++ E Qt

Pré-requisitos

- ◆ Lógica de programação
- ◆ Fundamentos QML
- ◆ Fundamentos de JavaScript (desejável)

Carga Horária

40h

Método

- ◆ Fundamentação expositiva
- ◆ Práticas em laboratório

Conteúdo

- ◆ Introdução às arquiteturas RESTful
- ◆ Servidores RESTful com o Ruby Sinatra
- ◆ Clientes RESTful com QML e JavaScript
- ◆ Clientes RESTful com QML e C++
- ◆ Trabalhando com *caches* locais
- ◆ Operações desconectadas e sincronização
- ◆ Funcionalidades básicas para *web scraping*
- ◆ Arquiteturas de clientes RESTful com QML

DESENVOLVENDO APLICAÇÕES ANDROID COM Qt

Visão Geral e Objetivos

Aplicativos para plataformas móveis, tais como *smartphones* e *tablets*, têm modificado drasticamente a forma com que interagimos com a tecnologia. Neste cenário, o Qt apresenta a particular vantagem de suportar plataformas tais como Android e iOS com um código-fonte único, reduzindo custos e facilitando a manutenção, ao mesmo tempo em que apresenta excelente desempenho das aplicações.

Neste treinamento, são apresentados os principais recursos que o Qt disponibiliza para implementação de aplicativos para a plataforma Android. Ao final do treinamento, o aluno deverá ser capaz de criar aplicativos que utilizam recursos tais como câmera, sensores, geolocalização e multimídia.

Pré-requisitos

- ◆ Lógica de programação
- ◆ Fundamentos QML
- ◆ Fundamentos de JavaScript (desejável)

Carga Horária

40h

Método

- ◆ Fundamentação expositiva
- ◆ Práticas em laboratório

Conteúdo

- ◆ Visão geral do Qt para Android
- ◆ QML em dispositivos móveis
- ◆ Suporte a *touch* e *multitouch*
- ◆ Acesso à câmera
- ◆ Geolocalização e serviços de mapas
- ◆ Utilizando sensores
- ◆ *Bluetooth* e *NFC*
- ◆ Efeitos gráficos e sistemas de partículas
- ◆ Áudio e vídeo
- ◆ Integrando código Java
- ◆ Implementando *Android Services* com Qt
- ◆ Lidando com permissões
- ◆ Geração de *apks* e pacotes AAB
- ◆ Lidando com SSH
- ◆ Publicando em lojas de aplicativos



PADRÕES DE PROJETO COM Qt

Visão Geral e Objetivos

A arquitetura de um software define aspectos básicos em relação à sua estrutura e comportamento. Arquiteturas bem projetadas são fundamentais para melhorar a manutenibilidade, adiar o envelhecimento do software e viabilizar o atendimento de outros requisitos não-funcionais. Padrões de projeto representam soluções de qualidade para problemas recorrentes durante a fase de projeto detalhado de arquiteturas.

Neste treinamento, são apresentados os principais estilos/padrões arquiteturais utilizados em aplicações Qt, os padrões de projeto já disponibilizados no *toolkit* e como novos padrões podem ser implementados. Ao final do treinamento, o aluno deverá ser capaz de projetar e implementar arquiteturas de software efetivas para aplicações Qt.

Pré-requisitos

- ◆ Fundamentos de C++
- ◆ Desenvolvendo aplicações gráficas com QtWidgets



ESPECIALISTAS EM C++ E Qt

Carga Horária

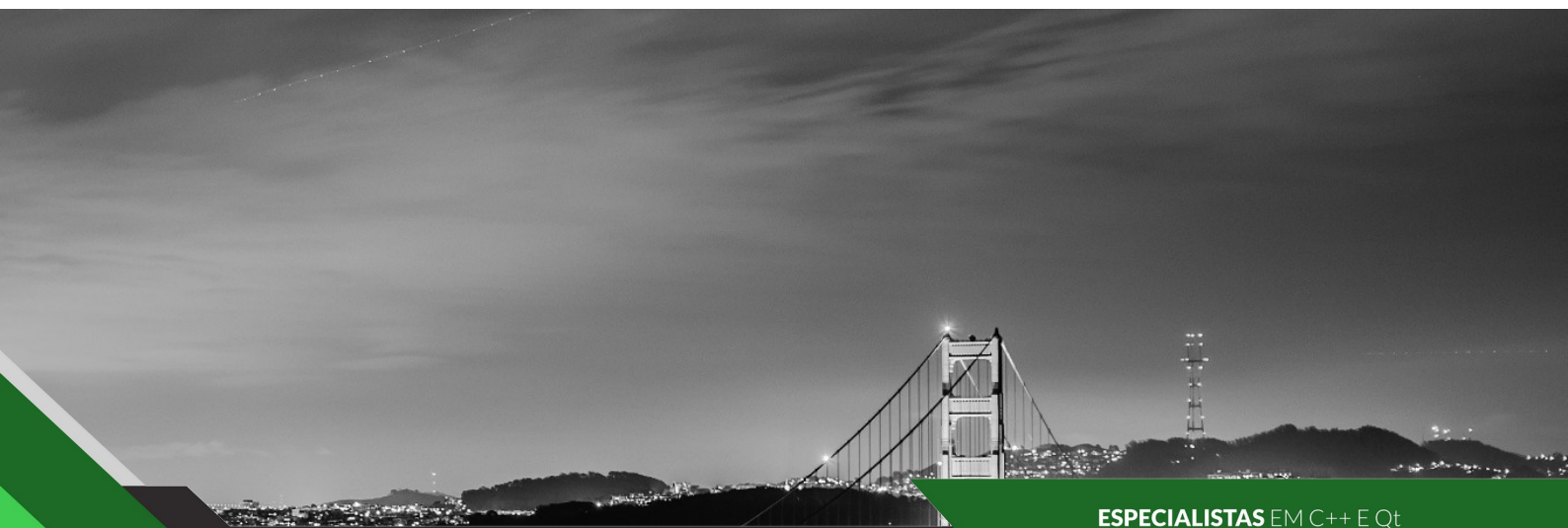
40h

Método

- ◆ Fundamentação expositiva
- ◆ Práticas em laboratório

Conteúdo

- ◆ Fundamentos de arquitetura de software
- ◆ Arquiteturas em *microkernel*
- ◆ Trabalhando com *plug-ins*
- ◆ Construindo SDKs com base no Qt Creator
- ◆ *Abstract Factory* e *Factory Method*
- ◆ *Composite*, *Bridge*, *Decorator* e *Adapter*
- ◆ *Template Method*, *Observer* e *Command*
- ◆ *Strategy*, *Flyweight* e *Visitor*
- ◆ *MapReduce* e *Future*
- ◆ Idiomas de programação para C++ e Qt



COMPUTAÇÃO GRÁFICA COM Qt 3D

Visão Geral e Objetivos

A Computação Gráfica está atualmente presente em uma série de áreas, desde os jogos e *displays* digitais até o uso no cinema, na realidade aumentada e na visualização científica. A utilização de frameworks produtivos, com bom desempenho e com abstrações satisfatórias é fundamental neste domínio de aplicação.

Neste treinamento, são apresentados os principais recursos do Qt 3D: um grupo de módulos do Qt para construção de aplicações voltadas à modelagem e renderização de cenas 2D e 3D. Ao final do treinamento, o aluno deverá ser capaz de criar aplicações Qt que realizam a visualização de malhas poligonais, bem como o uso de recursos ligados a textura e animações.



ESPECIALISTAS EM C++ E Qt

<http://qmob.solutions>

Pré-requisitos

- ◆ Fundamentos de C++
- ◆ Desenvolvendo aplicações gráficas com QtWidgets
- ◆ Fundamentos de QML

Carga Horária

40h

Método

- ◆ Fundamentação expositiva
- ◆ Práticas em laboratório

Conteúdo

- ◆ Introdução ao Qt 3D
- ◆ Malhas poligonais e geometrias
- ◆ Materiais
- ◆ *Shaders*
- ◆ Mapeamento de sombra
- ◆ Oclusão de ambiente
- ◆ Simulação de física
- ◆ Detecção de colisão
- ◆ Corpos rígidos
- ◆ Partículas
- ◆ Técnicas para construção de animações

SISTEMAS EMBARCADOS COM Qt

Visão Geral e Objetivos

Os sistemas embarcados estão atualmente presentes em diversas áreas tais como medicina, agronegócio, indústria automotiva, eletrônica de consumo e entretenimento. Desenvolver soluções de software que funcionem de maneira satisfatória em plataformas com restrições de memória, processamento e comunicação é uma tarefa desafiadora.

Neste treinamento, são apresentados os conceitos básicos para construção de imagens Linux com suporte a Qt, utilizando as tecnologias *buildroot* e *Yocto*. Tecnologias como *Boot2Qt* e *Qt Safe Renderer* são discutidas e os principais recursos do Qt para interação com sensores e atuadores são apresentados. Ao final do treinamento, o aluno deverá ser capaz de projetar e implementar soluções embarcadas com base no Qt.

Pré-requisitos

- ◆ Fundamentos de Linux
- ◆ Lógica de programação

Carga Horária

32h

Método

- ◆ Fundamentação expositiva
- ◆ Práticas em laboratório

Conteúdo

- ◆ Fundamentos de Linux embarcado
- ◆ *Toolchains*, *bootloaders* e compilação cruzada
- ◆ Plug-ins de renderização Qt em sistemas embarcados
- ◆ Construindo imagens Linux embarcado com *buildroot*
- ◆ Construindo imagens Linux embarcado com *Yocto*
- ◆ Interagindo com periféricos de propósito geral
- ◆ Desenvolvendo em Qt para o Raspberry Pi
- ◆ Desenvolvendo em Qt para a Toradex Colibri i.MX 8
- ◆ Otimizações para baixo consumo de energia
- ◆ Bluetooth LE
- ◆ Padrões arquiteturais para sistemas embarcados



ESPECIALISTAS EM C++ E Qt

<http://qmob.solutions>

OTIMIZANDO APLICAÇÕES Qt

Visão Geral e Objetivos

Embora o Qt seja um *toolkit* projetado para viabilizar o desenvolvimento de sistemas de alto desempenho, alguns cuidados são necessários ao trabalhar em domínios como computação de alto desempenho, *data-intensive systems* ou sistemas embarcados. Nestes casos, utilizar corretamente os recursos oferecidos pelo Qt é fundamental para atender os requisitos não-funcionais envolvidos.

Neste treinamento, são discutidos pontos críticos onde geralmente aplicações Qt perdem desempenho, as principais ferramentas de *debugging* e *profiling* utilizadas e diretrizes básicas para otimização de aplicações Qt. Ao final do treinamento, o aluno deverá ser capaz de realizar análises de aplicações Qt já existentes e refatorações para melhoria do desempenho.

Pré-requisitos

- ◆ Desenvolvendo aplicações gráficas com QtWidgets
- ◆ Fundamentos de QML

Carga Horária

32h



ESPECIALISTAS EM C++ E Qt

<http://qmob.solutions>

Método

- ◆ Fundamentação expositiva
- ◆ Práticas em laboratório

Conteúdo

- ◆ *Debugging e profiling* de aplicações Qt
- ◆ Armadilhas de desempenho no Qt
- ◆ GDB, *AddressSanitizer*, *ThreadSanitizer* e *Gamma Ray*
- ◆ Analisadores estáticos de código
- ◆ *Linux perf*, *hotspot* e *heaptrack*
- ◆ Otimizando aplicações QML com o *QML Profiler*
- ◆ *Valgrind* e *Massif Visualizer*
- ◆ *KCachegrind*
- ◆ Otimizando a inicialização de aplicações Qt
- ◆ Otimizando *footprint*



C++ AVANÇADO

Visão Geral e Objetivos

O C++ é uma linguagem multiparadigma, o que a faz adequada para uso em uma variedade de cenários e domínios de aplicação. Recursos avançados tais como *templates*, metaprogramação, *Standard Template Library*, RTTI e as melhorias introduzidas no C++11, 14, 17 e 20 são importantes para construir sistemas de melhor qualidade e para resolver problemas que requerem estratégias não tão comuns na orientação a objetos.

Neste treinamento, são apresentados os recursos do C++ que contribuem para o desenvolvimento de sistemas mais flexíveis e robustos. Ao final de treinamento, o aluno deverá ser capaz de dominar técnicas tais como *p-impl*, *templates*, metaprogramação, *smart pointers*, *constexpr*, *perfect forwarding* e *lambda expressions*.



Pré-requisitos

- ◆ Fundamentos de C++

Carga Horária

40h

Método

- ◆ Fundamentação expositiva
- ◆ Práticas em laboratório

Conteúdo

- ◆ Compatibilidade binária e *p-impl*
- ◆ *Templates* e programação genérica
- ◆ Fundamentos de metaprogramação
- ◆ *Standard Template Library* (STL)
- ◆ *Smart pointers*
- ◆ *constexpr*
- ◆ Inferência de tipos (*auto* e *decltype*)
- ◆ *Move semantics* e *Perfect forwarding*
- ◆ *Lambda expressions*

QML AVANÇADO

Visão Geral e Objetivos

O QML tem sido, já há algum tempo, a linguagem de preferência para o desenvolvimento de interfaces gráficas baseadas em Qt. Recursos avançados, tais como gerenciamento dinâmico de elementos, integração com C++, customização através de estilos e WebSockets fazem com que aplicações QML sejam atualmente desenvolvidas para áreas tais como IoT, sistemas automotivos e aeroespaciais, dispositivos médicos, dentre outras.

Neste treinamento, são apresentados recursos avançados que viabilizam a integração sinérgica entre o QML e o C++, bem como funcionalidades para gerenciamento dinâmico de elementos QML. Ao final do treinamento, o aluno deverá ser capaz de projetar e implementar aplicações híbridas QML/C++ de forma produtiva.

Pré-requisitos

- ◆ Fundamentos de C++
- ◆ Fundamentos de QML

Carga Horária

40h

Método

- ◆ Fundamentação expositiva
- ◆ Práticas em laboratório

Conteúdo

- ◆ Gerenciamento dinâmico de objetos QML (*Repeaters, Loaders e components*)
- ◆ Integração QML/C++ (aspectos avançados):
 - ◆ Motivação para integração QML/C++
 - ◆ Tipos de integração QML/C++
 - ◆ Definindo e registrando tipos QML em C++
 - ◆ Interagindo com objetos QML via C++
- ◆ Integração QML/openGL
- ◆ *User Input* (aspectos avançados): *input focus, virtual keyboard e multi-touch*
- ◆ Plug-ins QML
- ◆ *WebChannel* e *WebSocket* com QML
- ◆ Manipulando elementos gráficos no QML
- ◆ Animações avançadas
- ◆ Testes unitários com QML
- ◆ Configurando o *look'n'feel* de aplicações QML com estilos



GERÊNCIA DE QUALIDADE, DEVOPS E CONTINUOUS DELIVERY PARA PROJETOS Qt

Visão Geral e Objetivos

Os sistemas de software atuais são complexos o suficiente para fazer com que o uso de ferramentas básicas tais como IDE's e depuradores não sejam mais suficientes para entregar sistemas de qualidade. Práticas atuais da Engenharia de Software, tais como DevOps e *Continuous Delivery* (CD) se tornaram indispensáveis para o desenvolvimento de soluções de sucesso.

Neste treinamento, são apresentadas as principais tecnologias e processos responsáveis pela gerência de qualidade em projetos Qt. Isto inclui, por exemplo, a utilização de modelos de *branching* adequados, *Continuous Integration* (CI), *Continuous Delivery* (CD), políticas para testes e revisões, bem como ferramentas para gerência de projeto. Ao final do treinamento, o aluno deverá ser capaz de projetar e implantar uma política de gerência de qualidade adequada para a realidade de projetos Qt em uma determinada empresa.

Pré-requisitos

- ◆ Desenvolvendo aplicações gráficas com QtWidgets
- ◆ Fundamentos de QML
- ◆ Noções de gerência de projetos (desejável)



Carga Horária

32h

Método

- ◆ Fundamentação expositiva
- ◆ Práticas em laboratório

Conteúdo

- ◆ Introdução à Gerência de Qualidade
- ◆ Git e modelos de *branching*
- ◆ O *pipeline* de qualidade
- ◆ Qt e testes automatizados
- ◆ Testes de sanidade
- ◆ Analisadores estáticos de código
- ◆ Qt e GitLab (*merge requests*, *reviews* e *pipelines*)
- ◆ *Continuous Integration* e *Continuous Delivery* com Qt e GitLab
- ◆ *Continuous Deployment* com Qt e Docker
- ◆ *Release management*
- ◆ Gerência de qualidade e métodos ágeis



Qt FOR PYTHON

Visão Geral e Objetivos

A linguagem Python é atualmente uma das tecnologias mais utilizadas no desenvolvimento de sistemas em diversos domínios e para múltiplas plataformas. Características tais como excelente produtividade, abstrações de alto nível, tipagem dinâmica, suporte a múltiplos paradigmas e a disponibilidade de uma variedade de bibliotecas para diversos propósitos são alguns dos principais fatores que contribuem para a ampla adoção atual do Python na indústria.

Este treinamento apresenta os fundamentos do Qt for Python – *binding* oficial para utilização do Qt em aplicações Python. São abordados tópicos tais como geração de *bindings* via Shiboken, módulos Qt disponíveis, integração com Qt Designer e Qt Creator, bem como aspectos de implantação.

Pré-requisitos

- ◆ Lógica de programação
- ◆ Noções de Python (desejável)



ESPECIALISTAS EM C++ E Qt

Carga Horária

40h

Método

- ◆ Fundamentação expositiva
- ◆ Práticas em laboratório

Conteúdo

- ◆ Motivação para o Qt for Python
- ◆ Histórico dos *bindings* Python do Qt
- ◆ Criando uma aplicação QtWidgets
- ◆ Sinais e *slots*
- ◆ *Layouts*
- ◆ Criando uma aplicação QML/QtQuick
- ◆ Integração com o Qt Designer
- ◆ Visualização de dados no Qt for Python
- ◆ Trabalhando com multimídia
- ◆ Integração de conteúdo web com *WebEngine*
- ◆ Integração com C++
- ◆ Gerando novos *bindings* com o Shiboken
- ◆ Aspectos de implantação (com *fbs*, *PyInstaller* e *cx_Freeze*)
- ◆ Integração com bibliotecas de terceiros

DESENVOLVIMENTO DE JOGOS COM Qt

Visão Geral e Objetivos

A indústria de jogos é um dos setores mais dinâmicos, promissores e desafiadores da indústria atual. Desenvolver soluções com alto desempenho, alta flexibilidade, UIs modernas e com *time-to-market* aceitável é um desafio enfrentado por todas as empresas da área. Adicionalmente, projetar e desenvolver soluções que funcionem nas múltiplas plataformas de hardware e de comunicação atualmente disponíveis torna a situação ainda mais difícil.

Este treinamento apresenta os fundamentos para construção de jogos 2D e 3D utilizando o Qt. Serão discutidos os principais aspectos relacionados à renderização de elementos 2D e 3D, detecção de colisões, múltiplas técnicas de animação, arquiteturas ECS (*Entity-Component-System*), funcionalidades para multimídia e aspectos de comunicação em rede. Ao final do treinamento, o aluno deverá ser capaz de projetar e implementar jogos 2D e 3D em Qt.

Pré-requisitos

- ◆ Lógica de programação
- ◆ Noções de Computação Gráfica (desejável)
- ◆ Computação Gráfica com Qt 3D (desejável)

Carga Horária

40h

Método

- ◆ Fundamentação expositiva
- ◆ Práticas em laboratório

Conteúdo

- ◆ Desenvolvendo jogos 2D com QtWidgets e *Qt Graphics View Framework*
 - ◆ Sistemas de coordenadas e criação/manipulação de itens gráficos
 - ◆ Animando propriedades com *easing curves* e detectando colisões
- ◆ Desenvolvendo jogos 2D com QML/QtQuick
 - ◆ Posicionando objetos QML
 - ◆ *Animando propriedades com easing curves e detectando colisões*
- ◆ Desenvolvendo jogos 3D com o Qt 3D
 - ◆ Principais recursos das APIs C++ e QML
 - ◆ Malhas poligonais, texturas, mapeamentos e detecção de colisão
- ◆ Adicionando recursos multimídia
- ◆ Fundamentos para construção de jogos em rede com Qt
- ◆ Jogos para Android e iOS



PROJETO E ANÁLISE ARQUITETURAL

Visão Geral e Objetivos

Entregar sistemas de software de qualidade é uma atividade que envolve a adoção de técnicas e ferramentas adequadas, desde a análise de requisitos até a implantação e evolução do sistema. Uma atividade importante neste processo é o projeto e análise da arquitetura do software. É a arquitetura do software que viabiliza o atendimento dos principais requisitos não-funcionais e faz com que o software se mantenha útil e modificável ao longo de vários anos.

Nestas atividades de consultoria, realizamos o projeto de arquiteturas para novos sistemas ou a análise de arquiteturas já projetadas, com o objetivo de identificar pontos de mal formação ou passíveis de melhoria.

Atividades

Projeto arquitetural; modelagem de visões arquiteturais; definição de restrições arquiteturais; projeto detalhado e definição de tecnologias; análise arquitetural; integração arquitetura ↔ gerência de qualidade; técnicas generativas; arquiteturas em processos ágeis.



PROCESSOS DE DESENVOLVIMENTO E GERÊNCIA DE QUALIDADE

Visão Geral e Objetivos

Desenvolver software de qualidade requer a implantação de processos maduros de desenvolvimento, onde institucionaliza-se o cumprimento de etapas e os produtos gerados. O objetivo é sistematizar atividades e adotar técnicas e ferramentas que reduzem a ocorrência de *bugs* e degradações arquiteturais.

Nestas atividades de consultoria, realizamos um mapeamento das características do seu negócio e construímos um plano de implantação de processo de desenvolvimento e gerência de qualidade, particularmente adaptados à realidade da sua empresa.

Atividades

Caracterização do negócio; determinação do grau de agilidade requerido; definição do processo de desenvolvimento; definição do processo de gerência de qualidade; seleção de práticas e ferramentas; plano de implantação; treinamento de equipe.



PLANEJAMENTO DE MIGRAÇÃO PARA Qt

Visão Geral e Objetivos

O uso de sistemas legados, com restrições e/ou obsolescências relacionadas a arquiteturas, linguagens de programação, interfaces gráficas de usuário e capacidades de integração, é uma situação bastante comum em muitas corporações. Migrar tais soluções para tecnologias mais poderosas e flexíveis deve ser sempre considerado, com o objetivo de adiar o envelhecimento do software. Nestas atividades de consultoria, analisamos a solução atualmente adotada na sua empresa e especificamos um plano de migração com foco na adoção do Qt e de outras tecnologias relacionadas. O objetivo é realizar um levantamento preliminar de custos, prazos e equipe necessária para implementar a nova solução.

Atividades

Análise da arquitetura da solução atual; análise de requisitos funcionais e não-funcionais; avaliação de necessidade de reprojeto arquitetural; definição de tecnologias necessárias; planejamento da migração; estimativa de custos, prazos e equipe.



ANÁLISE DE LICENCIAMENTO Qt

Visão Geral e Objetivos

O Qt é atualmente licenciado em duas modalidades: Qt Open Source (licenciamento gratuito) e Qt Comercial (licenciamento pago). O uso do Qt Open Source é regulado através da licença (L)GPL v3, que traz uma série de benefícios e obrigações a serem atendidas. Em alguns cenários, domínios de aplicação e modelos de comercialização, entretanto, certos aspectos da (L)GPL v3 não podem ser atendidos e, portanto, é necessário que uma licença comercial seja adquirida.

Este grupo de atividades tem como objetivo analisar o cenário técnico, comercial e estratégico do cliente, de modo a indicar se a (L)GPL v3 pode ser integralmente atendida ou se há demandas para uso de licença comercial.

Atividades

Levantamento do tipo de produto sendo desenvolvido (aplicação, solução embarcada, SDK, etc); análise dos requisitos da LGPL v3; considerações sobre linkagem estática x dinâmica; tipos de licença.



DESENVOLVIMENTO OUTSOURCE DE SOFTWARE

Visão Geral e Objetivos

A **qmob** solutions conta com uma equipe completa de desenvolvedores e arquitetos Qt para realizar a construção do seu sistema *desktop*, *mobile* ou embarcado. Além do desenvolvimento das soluções de *front-end*, contamos com profissionais experientes no desenvolvimento de *back-ends* RESTful utilizando tecnologias tais como Sinatra (Ruby) e Flask (Python). Com isso, a **qmob** solutions entrega soluções completas, garantindo a adoção das melhores práticas da Engenharia de Software e a entrega de sistemas bem arquitetados e fáceis de evoluir.

Pré-requisitos

- ◆ Levantamento/Documento de requisitos
- ◆ Planejamento do projeto de desenvolvimento

Carga Horária

Variável



ESPECIALISTAS EM C++ E Qt

<http://qmob.solutions>

Método

- ◆ Métodos ágeis
- ◆ *Lean Architecture*

Atividades

- ◆ Levantamento de requisitos
- ◆ Projeto arquitetural
- ◆ Projeto detalhado
- ◆ Planejamento de *sprints*
- ◆ Desenvolvimento do sistema
- ◆ Definição de estratégias de teste
- ◆ Definição de estratégias de *Continuous Integration* (CI)
- ◆ Definição de estratégias de *Continuous Delivery* (CD)
- ◆ Processo de *release*
- ◆ Planejamento de implantação
- ◆ Planejamento de evolução





TREINAMENTOS, CONSULTORIA
& DESENVOLVIMENTO

2020