**Relatório**

**Prova de Aptidão Profissional**

**Escola Secundária de Estarreja**

Curso Profissional – Técnico de Gestão e Programação de Sistemas Informáticos



**Formando:** Gabriel Marques Castanheira Valente

Prova de Aptidão Profissional Realizada no âmbito do curso

Profissional Técnico de Gestão e Programação de Sistemas Informáticos

Para a prova de aptidão profissional decidi desenvolver uma aplicação em C# e XAML, em que o modelo é WPF (Windows Presentation Foundation) que simplificando é uma aplicação onde em vez de utilizar apenas uma linguagem dedicada para a parte lógica, utiliza outra linguagem para parte gráfica. Este modelo de aplicação trás vantagens como performance entre outros aspetos.

A aplicação que desenvolvi tem o nome de “**WorkPlanner**” e consiste em uma aplicação de gestão de trabalhos. Para alguém que trabalha por conta própria e precisa de uma aplicação para tomar conta das horas que faz para cada trabalho, esta é a aplicação que permite manter o registo de tudo.

AGRADECIMENTOS

Durante a realização desta prova pude melhorar todos os meus conhecimentos em C# e aprender uma linguagem nova que foi XAML. A integração de SQL na aplicação tambem foi muito desafiadora. E para a boa realização desta prova pude contar com o apoio dos familiares, assim como de colegas e amigos, que estiveram sempre dispostos a ajudar.

Com certeza que tenho que agradecer aos familiares por estarem sempre dispostos a ajudar e a dar opiniões em relação a vários pontos, assim como tenho de agradecer a colegas e amigos por testarem a aplicação ou até mesmo por me ajudarem a corrigir problemas, e tambem na planificação de alguns pontos que na altura não estava a conseguir entender como os resolver.

Índice

[Introdução 3](#_Toc12996276)

[Fundamentação da escolha 4](#_Toc12996277)

[WorkPlanner 5](#_Toc12996278)

[Instalação 5](#_Toc12996279)

[Abertura 7](#_Toc12996280)

[Código da página 9](#_Toc12996281)

[Página Principal 10](#_Toc12996282)

[Código da página 11](#_Toc12996283)

[Gerir Clientes 14](#_Toc12996284)

[Código da página 16](#_Toc12996285)

[Gerir Serviços 20](#_Toc12996286)

[Código da página 20](#_Toc12996287)

[Gerir Trabalhos 21](#_Toc12996288)

[Código da página 22](#_Toc12996289)

[Gerir Tarefas 24](#_Toc12996290)

[Código da página 25](#_Toc12996291)

[Definições 26](#_Toc12996292)

[Código da página 27](#_Toc12996293)

[Conclusão 28](#_Toc12996294)

[Webgrafia 29](#_Toc12996295)

# Introdução

Para esta prova, foram utilizadas as linguagens C#, XAML e SQL, para a parte logica, gráfica e de base de dados, respetivamente. O desenvolvimento desta aplicação trouxe muitas dificuldades que fui capaz de superar.

Como anteriormente foi dito esta aplicação consiste em um organizador de trabalhos, é possível guardar várias informações dos clientes, dos serviços e dos trabalhos. Os clientes têm informações como nome, data de nascimento, morada, contactos. Os serviços têm o nome do serviço e o preço hora. Os trabalhos em si são dos mais complexos, sendo que cada trabalho tem um cliente, um nome e descrição, e é possível adicionar tarefas, que estas têm um serviço associado. As tarefas têm um serviço que vai ser utilizado o valor hora das mesmas, e para cada tarefa é possível adicionar quantas horas de trabalho forem necessárias. Todas estas contas vão ter ao trabalho que após tudo é possível calcular o valor do trabalho.

Este programa utiliza o modelo WPF que permite a placa gráfica carregar o visual do programa enquanto o CPU carrega a parte lógica do mesmo. Utilizei uma base de dados para manter o registo de todas estas informações. Decidi desenvolver a aplicação em C# pois já foi uma linguagem que trabalhei anteriormente, e gostava de melhorar ainda mais os meus conhecimentos com ela, e optei por WPF pois vim a saber das vantagens.

A linguagem XAML que significa E**x**tensible **A**pplication **M**arkup **L**anguage é uma linguagem parecia a HTML, no entanto esta é focada em aplicações, tanto de desktop como mobile. E é bastante flexível pois permite a edição tanto visualmente como por código.

# Fundamentação da escolha

Criei esta aplicação pois já tive a necessidade de um programa semelhante, onde possa saber o cliente e para cada trabalho possa apontar as horas. Antes de desenvolver recorria ao Excel com várias páginas para diferentes serviços, e entendi que não era da forma mais adequada nem a mais pratica. Por tanto como tenho mais conhecimentos e domino melhor algumas linguagens de programação do que Excel, achei que era a melhor oportunidade para poder melhorar a simples folha Excel que tinha.

Posso dizer que a aplicação está como eu imaginei, até com mais algumas funções que vim a achar que podiam melhorar o projeto num todo.

Considero que com o programa é muito mais simples de ter o registo de todos trabalhos, e que o mesmo está a fazer a sua função como planeei desde início.

# WorkPlanner

Para desenvolver este projeto utilizei C# na maioria do projeto, cerca de 6000 linhas, por toda a parte lógica do programa.

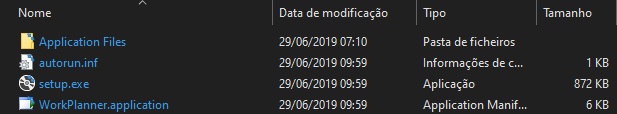
Para a parte gráfica utilizei XAML que como disse antes é semelhante a HTML, mas focado em aplicações, uma das grandes diferenças que encontrei entre ambos é que em HTML geralmente usa-se CSS para melhorar os elementos, em XAML são utilizadas tags como em HTML, mas todas as propriedades são escritas dentro do mesmo. E tambem é possível alterar essas mesmas propriedades dentro do próprio C# o que permite uma grande flexibilidade entre back e front.

E para guardar os dados utilizei um ficheiro de base de dados, em vez de utilizar um servidor, e para guardar algumas configurações utilizei o próprio sistema que está implementado neste tipo de aplicação. Utiliza um formato de XML, no entanto não é preciso ter nenhum conhecimento de como o mesmo funciona.

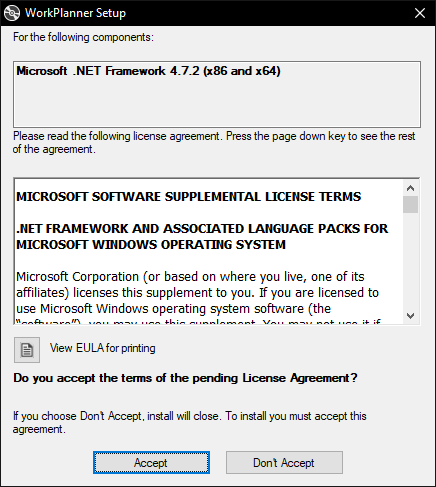
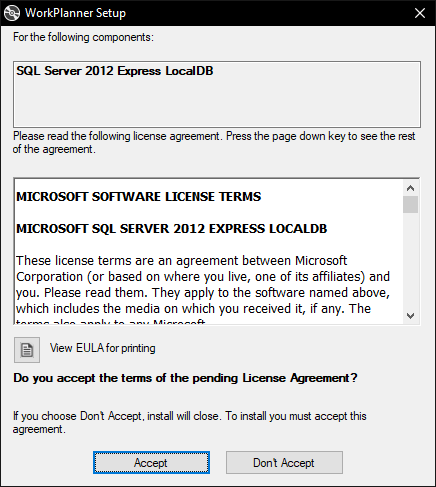
## Instalação

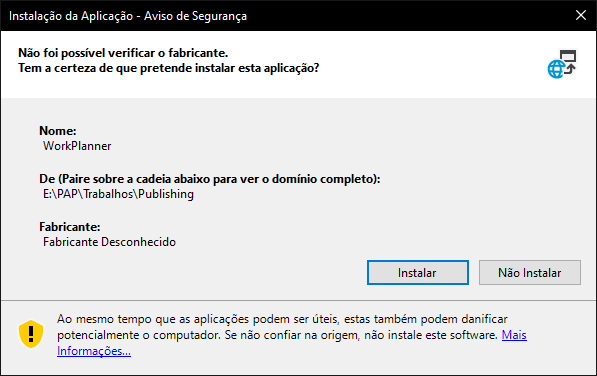
A aplicação para ser possível utilizar em qualquer computador é necessário instalar. Juntamente com o programa é instalado o “Microsoft .NET Framework 4.7.2 (x86 e x64)” e “SQL Server 2012 Express LocalDB”, assim como onze outros arquivos, entre eles partes de SQL, gráficos, base de dados entre outros.

A instalação é necessária pois cada computador é único e não é possível selecionar uma pasta diferente para cada um dos computadores, portanto, para a ligação á base de dados existe um comando que é “|DataDirectory|” que permite a própria aplicação escolher o local onde vai usar a base de dados.

Após descarregar a pasta com os ficheiros de instalação estamos presentes estes ficheiros.

A primeira pasta tem todos os arquivos que serão instalados, O “autorun.inf” serve caso o programa seja instalado por um CD/DVD que inicia automaticamente a instalação, que na entrega do projeto vou remover o mesmo. O “setup.exe” é o próprio instalador do programa que é semelhante a muitos outros instaladores. O último ficheiro são algumas informações como chave de publicação, dependências e entre outros parâmetros.

Após abrir o “setup.exe” deve aparecer as seguintes janelas, cada uma para instalar componentes diferentes, como é dito a cima.

Após isso irá aparecer dois UAC (User Account Control) que servem para permitir instalar os ficheiros necessários para o programa funcionar. De seguida aparecerá esta janela, para permitir instalar a aplicação.

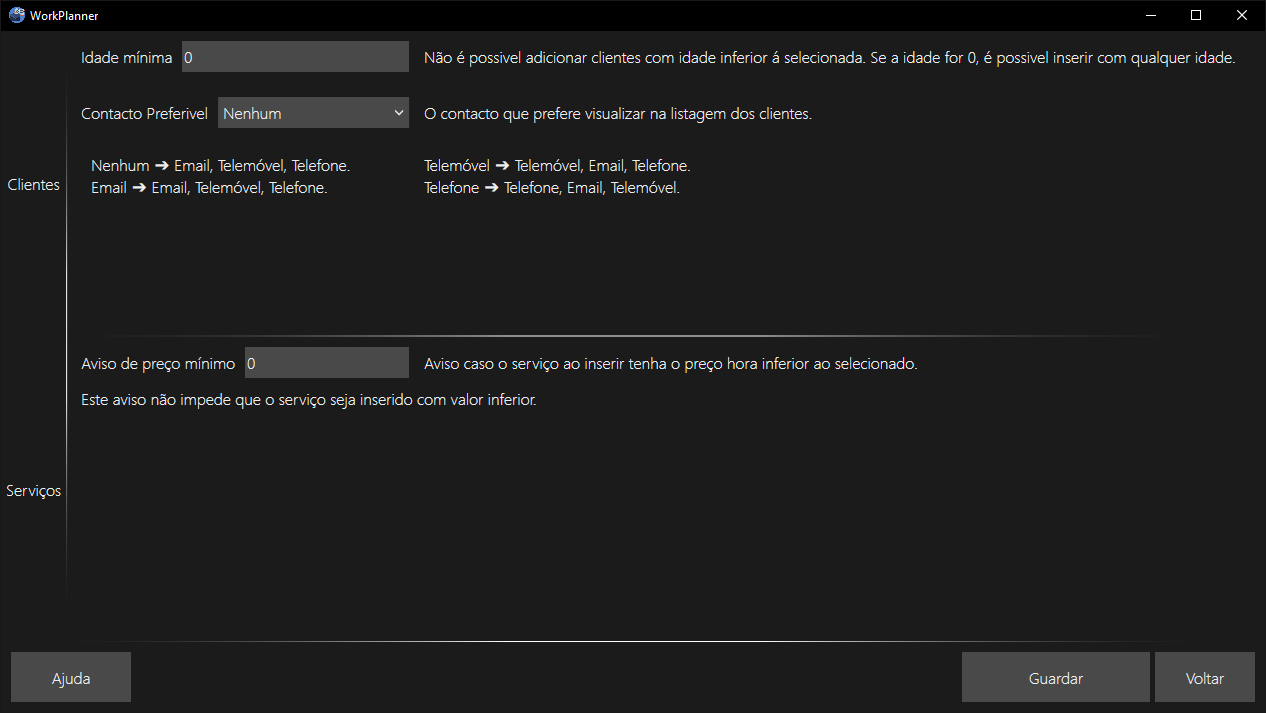
Após a instalação terminar um icon no ambiente de trabalho irá aparecer e o programa irá iniciar automaticamente.

## Abertura

Após o programa instalar ele será aberto automaticamente e irá aparecer um pequeno SplashScreen, enquanto é carregado, dependendo de computador para computador, não deverá demorar muito mais do que alguns segundos.

Esta é uma imagem simples apenas para mostrar que o programa está a iniciar. Depois de carregar, a primeira vez a entrar no programa aparecerá um género de Popup pois ainda não existem configurações guardadas, caso não seja a primeira vez e apareça, são problemas que podem acontecer se ao ler/escrever no ficheiro aconteça algo inesperado.

São presentes duas opções, sendo que é necessário editar algumas das configurações antes de iniciar por completo o programa.

Após carregar no botão de “Editar Configuração” somos direcionados para a página de definições onde existem 3 configurações.

A primeira é a idade mínima que o cliente tem de ter, isto pode ser em qualquer altura alterado, no entanto permite que no calendário esteja selecionado a partir de uma data.

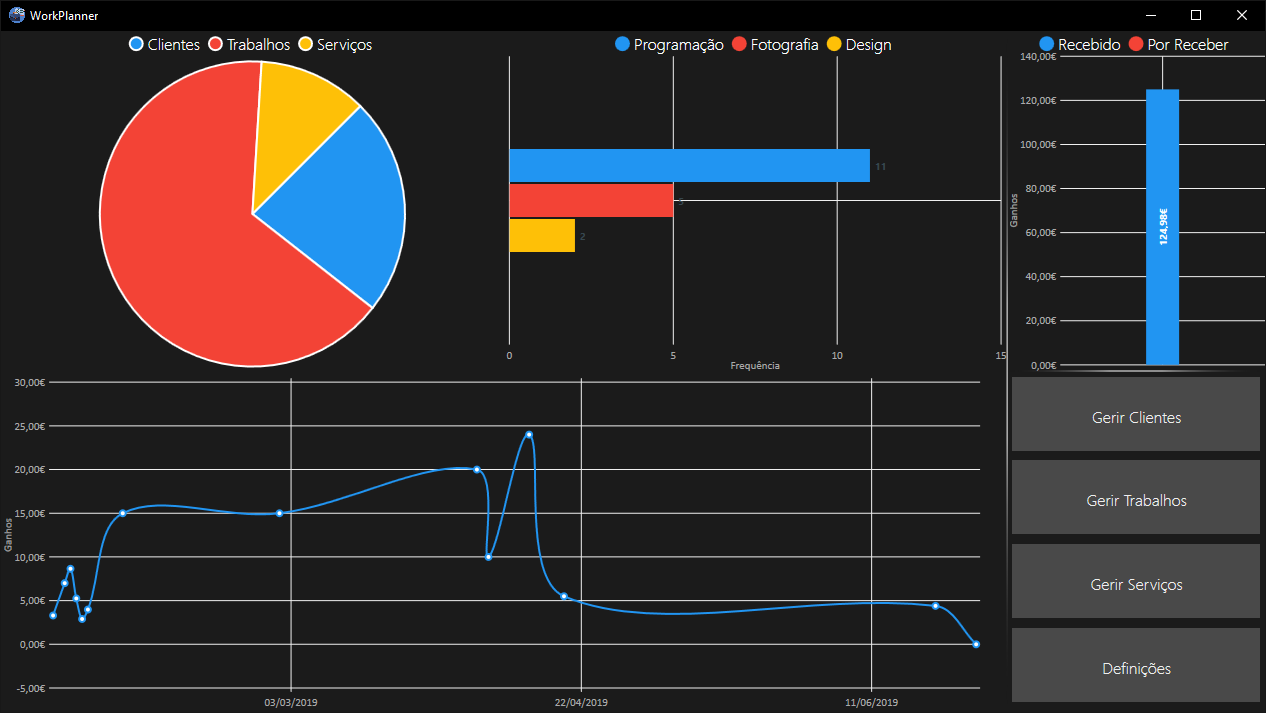
O contacto preferível é um campo que permite saber a ordem da listagem, ou seja, que na lista onde estão os clientes todos, dependendo da opção escolhida, se o cliente tiver os três contactos, a opção será a que ira mostrar.

Se o cliente tiver um email e um telemóvel, e a opção selecionada for “Email”, o email será o apresentado na listagem mas caso a opção seja “Telemóvel” o contacto apresentado será o telemóvel, mas como ele não tem telefone, e a opção selecionada for “Telefone” o contacto apresentado será o email, como está no esquema apresentado na imagem.

A terceira opção permite receber um aviso ao inserir um novo serviço que caso o preço hora seja inferior ao selecionado, iremos receber um aviso.

Muitas outras opções eram para ser inseridas, no entanto por causa de erros e de incompatibilidades não foi possível.

Mais á frente do relatório vou dar mais informações relativas a esta página.

Caso não seja a primeira vez a ligar o programa irá aparecer assim um menu. Esta é a página principal que só aparecem gráficos caso existam dados suficientes para os criar, caso contrário não será possível criar estes mesmos gráficos.

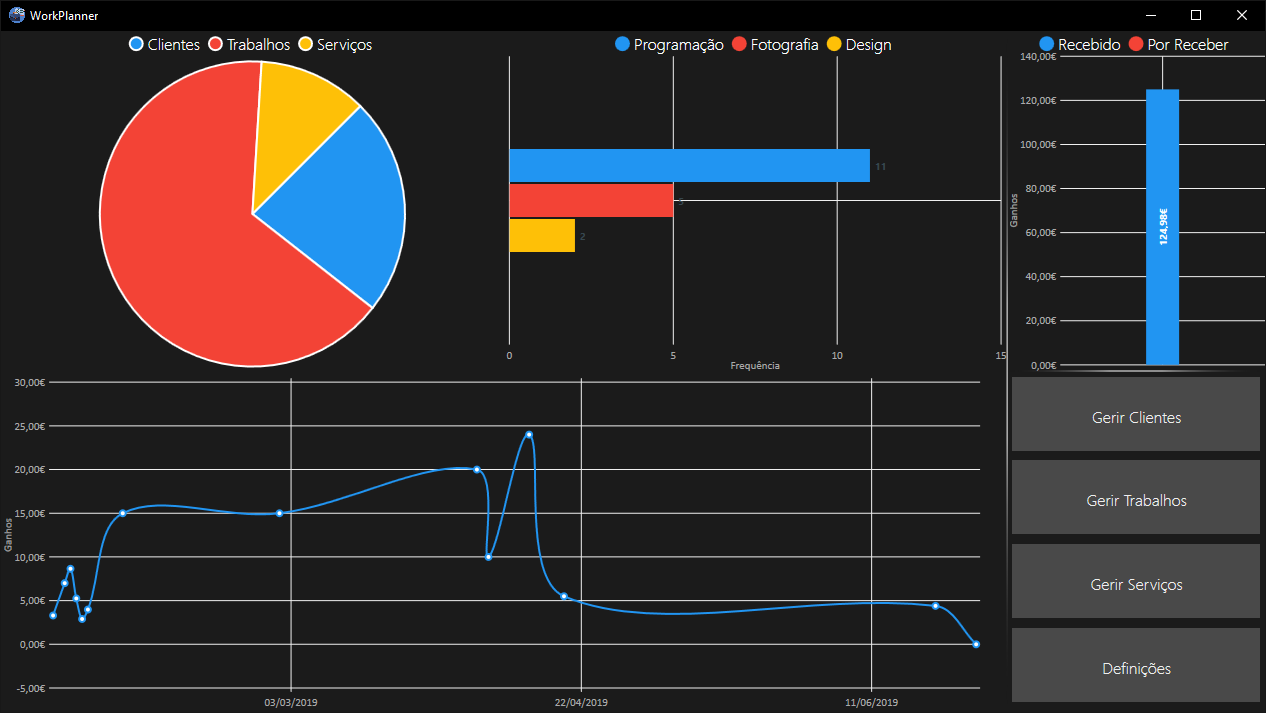
### Código da página

Esta página tem um simples esquema para descobrir se é a primeira vez ou não a iniciar a aplicação, no entanto funciona tambem caso não seja possível carregar os recursos.



Para explicar este parte de código é simples, ao iniciar a aplicação tenta ler as configurações, e caso não dê erro, este abre a página principal, a que se viu anteriormente, caso de algum erro (não existir uma configuração) irá aparecer o popup para se ir á página de configuração.

## Página Principal

A página principal tem um design simples, com quatro gráficos que achei mais pertinentes.

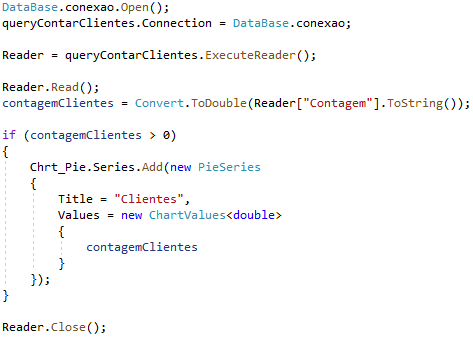
O gráfico circular são dos trabalhos, clientes e serviços onde se podem ver quais destes tem em maior quantidade. O seguinte, de barras horizontais, são o top 10 de serviços mais requisitados, e sendo que apenas estão guardados três serviços, apenas essas são mostrados. O outro gráfico de barras é a relação entre ganhos de todos os trabalhos e o que falta receber, por cima irá aparecer uma barra vermelha. O último gráfico que é de linhas, são os ganhos por dia, como existe uma tabela de tarefas é possível calcular por dia os ganhos, este é único gráfico que é possível fazer zoom no eixo do X, sendo que o eixo do Y é ajustado automaticamente.

Para alem dos quatro gráficos existe no canto inferior direito quatro botões que irei falar de seguida, mas cada um deles dá para uma página diferente.

O design desta aplicação foi inspirado nas aplicações da Adobe, com as cores correspondendo ao das aplicações, para alem das cores, não houve mais nenhuma outra aplicação em que me inspirei em termos de design.

### Código da página

Esta página consiste em código que é repetido de outras páginas pois a logica já estava correta, assim não é necessário reinventar a roda. Nesta página utilizei uma biblioteca que encontrei no GitHub e esta biblioteca é LiveCharts, que foi a que vi que “encaixava” melhor no tema do programa, todos outros que encontrei tinham um ar mais antiquado e não me pareciam estar bem documentados.



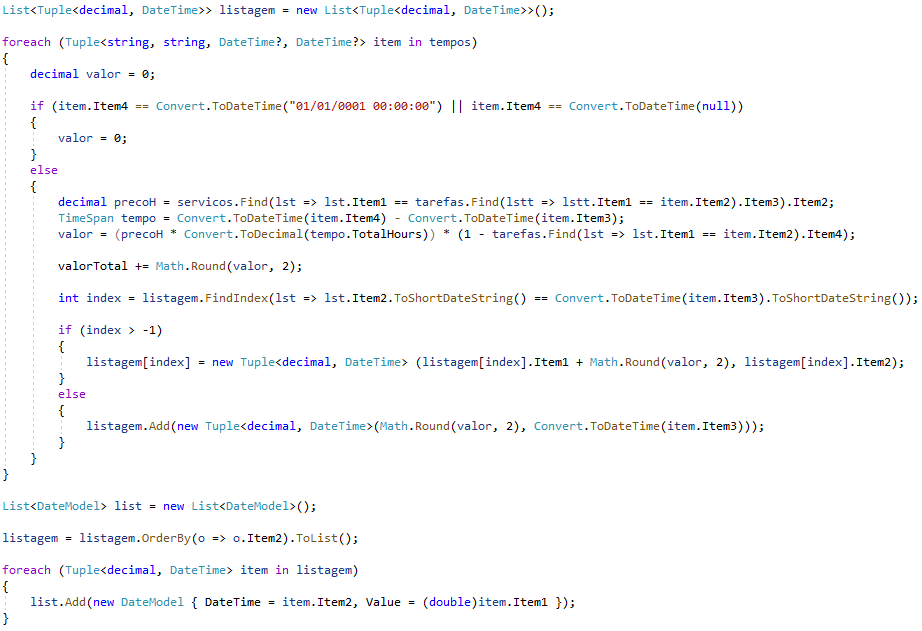
Este é um exemplo da inserção de um item no gráfico circular. Onde abro uma ligação com a base de dados, associo essa ligação á query que quero executar, e faço a query. Depois da query ser feita vou buscar o valor e coloco na variável da contagem dos clientes. Após isso eu adiciono esse valor ao gráfico, e dou o título de “Cliente”, neste caso, e adiciono o valor.

Isto acontece para os trabalhos e serviços. Este foi dos gráficos mais simples de adicionar valores, os seguintes houve algumas complicações.

Para o gráfico de linhas executei todo o tipo de queries para ter as informações de tudo, as queries foram:

* Chave e valor pago da tabela de trabalhos;
* Chave e preço hora da tabela de serviços;
* Chave de tarefas, chave de trabalho, chave de serviço e desconto da tabela tarefas;
* Chave de tempo, chave de tarefas, datas de início e fim da tabela de tempo.

Com estas informações foi possível calcular os ganhos por cada um dos dias, depois de ter recolhido todas estas informações tive de efetuar os cálculos por cada um dos dias.



O código em cima não é fácil de explicar, existe uma lista de todos os tempos na base de dados. Para cada item (registo independente), vou verificar se a data de fim não é nula (esta pode ser nula caso ainda não tenha esta tarefa).

Caso não seja nulo vou guardar o valor hora do serviço relativo a esse tempo, após isso preciso de saber o intervalo de tempo que trabalhou, para isso faço uma subtração entre a data de fim e a data de início. E por fim vou calcular o valor total, fazendo o preço hora a multiplicar pelas horas de trabalho isto é (tempo.totalHours, que vai buscar o tempo em horas, mesmo que seja um dia vou receber 24 horas), após estar calculado tal vai multiplicar pelo valor do desconto.

Apos estes cálculos existe uma variável que serve para receber os ganhos totais, este é para outro gráfico.

Depois a variável índex serve para verificar se já existe algum dia igual a esse, e assim adicionar os ganhos a esse dia, caso não encontre vai adicionar um novo registo á lista.

Este processo e feito para todos os tempos encontrados na base de dados. Após tal tarefa estar concluída os dados vão ser organizados, por ordem crescente. E após estar organizado os itens vão ser copiados para uma lista em que o sistema de gráficos entenda.

Outro dos gráficos é o de barras verticais que é o de comparação os ganhos dos trabalhos e o que falta pagar, geralmente deve-se ver apenas uma barra azul, e caso se veja uma barra azul e vermelha é porque ainda há trabalhos que faltam pagar, ou não foram totalmente pagos.

Este simplesmente tem a adição desses valores ao gráfico, tal como no gráfico circular. Sendo que todos os valores necessários foram guardados na leitura da base de dados anterior.



A imagem a cima é a query que tive de realizar para poder obter a contagem do top 10 de serviços mais requisitados.

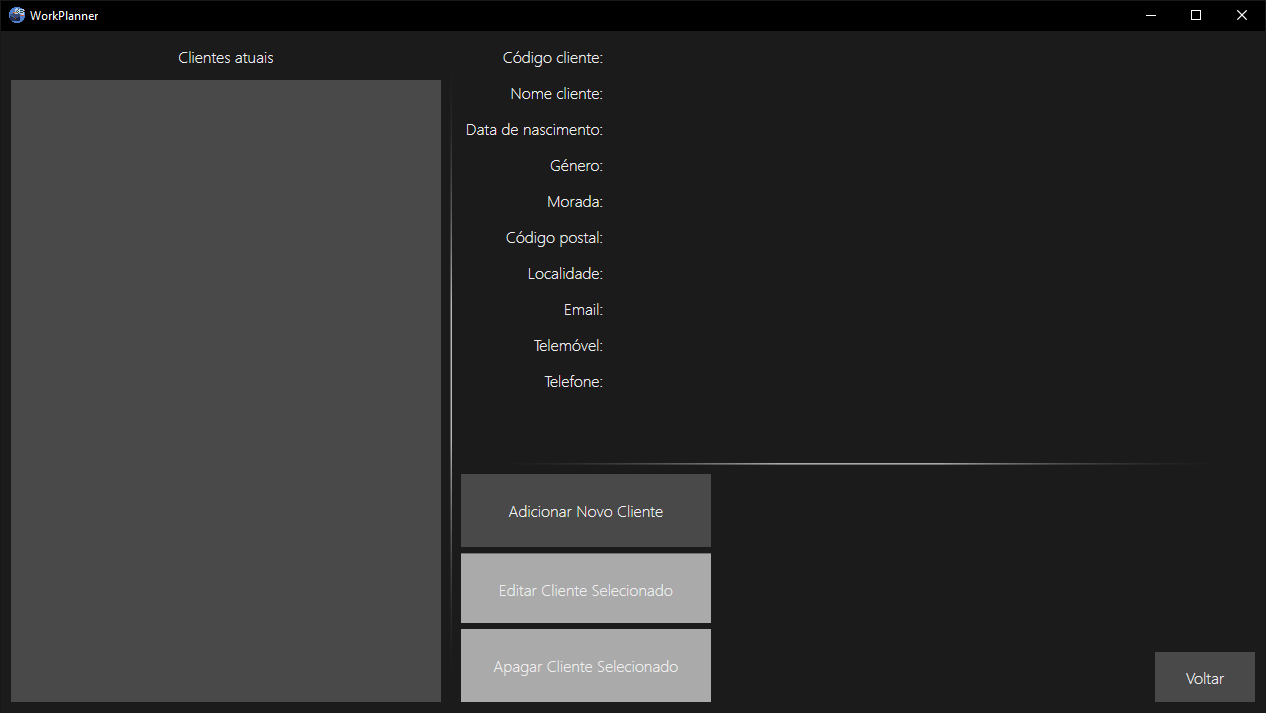


A adição destes valores tambem é simples, tal como em todos os outros gráficos.

Caso não seja possível ter os dados suficientes para a criação dos gráficos irá aparecer um aviso a dizer que não é possível.

## Gerir Clientes

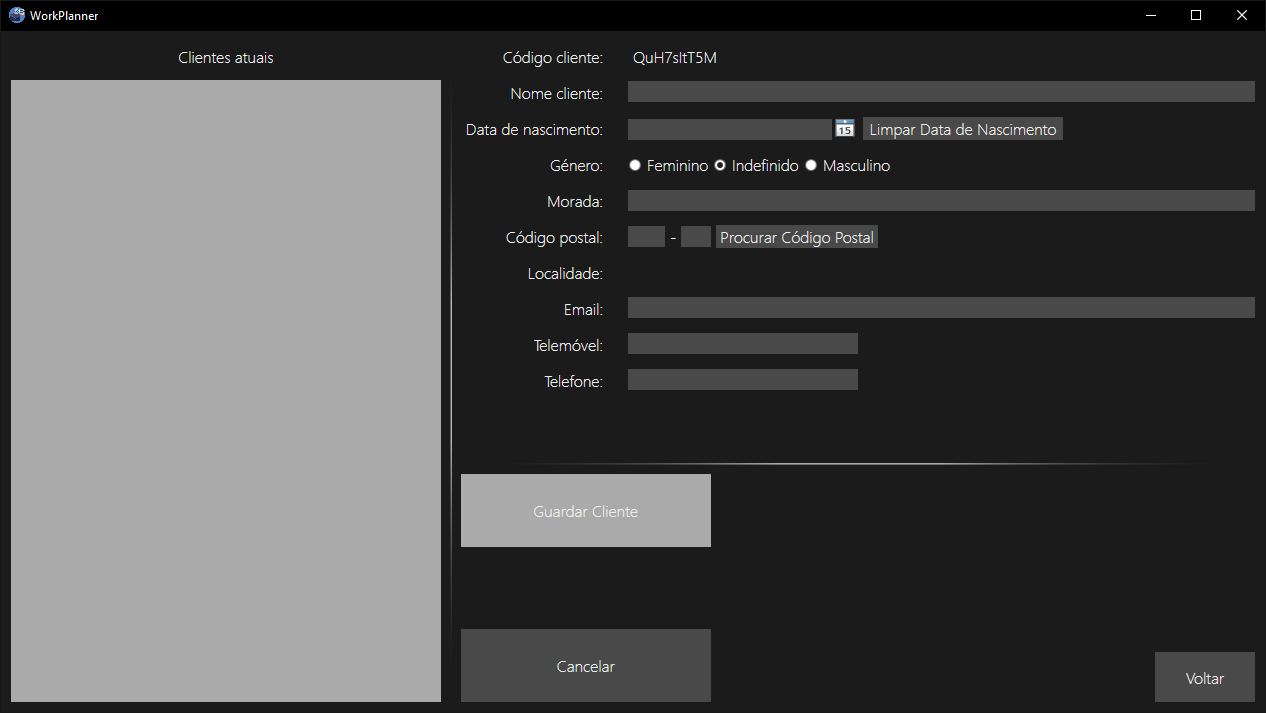
A página com mais campos para preencher, pois tem várias informações sobre o cliente sendo muitas delas não obrigatórias, as obrigatórias são, o nome (que tambem tem de ser único), e uma forma de contacto.



Esta página assim como todas as seguintes, com exceção da página das definições, tem o mesmo layout. Do lado esquerdo existe uma lista, que neste caso vão aparecer os clientes, e do lado direito temos a parte principal, em cima todos os campos, e em baixo as opções possíveis.

Os únicos campos obrigatórios são o nome, que tem de ser único, e é necessário ter uma forma de contacto. E o género, é obrigatório pois existem três botões de rádio, com masculino, indefinido e feminino.

A utilização do programa é simples, sendo que para adicionar um cliente novo é necessário carregar em “Adicionar Novo Cliente”.



Depois de entrar em modo de edição, é gerada uma chave de utilizador, esta é composta por 10 caracteres, entre eles, letras maiúsculas e minúsculas e números de 0 a 9. Os campos seguintes são validados em tempo real.

O nome de cliente não permite caracteres especiais, nem números, as primeiras letras do nome são sempre maiúsculas e as seguintes sempre minúsculas, esta validação permite caracteres com acentuação. Após sair da textbox o campo é validado com a base de dados para verificar que não existe nenhum nome igual, e caso não saia da textbox, ao fim de um segundo de ter escrito, é validado o campo.

A data de nascimento abre em vista de intervalo de anos, onde é possível selecionar o ano de nascimento do cliente, assim vai progredindo para o mês e para o dia de nascimento. Não é possivel inserir uma data inferior ao selecionado nas definições.

Os três campos seguintes funcionam em conjunto, sendo que é possível escrever o código postal e a localidade e a morada são automaticamente preenchidas. A morada só é preenchida caso não tenha texto na textbox, caso contrário apenas a localidade é preenchida. Caso não se carregue em “Procurar Código Postal” ao inserir estes campos serão inseridos á mesma corretamente.

Os últimos três campos são parcialmente obrigatórios, o que significa que apenas um deles é obrigatório, e todos estes têm validações.

Na edição do cliente, todos os campos são validados da mesma forma sendo possível colocar qualquer um dos campos sem valor caso seja necessário, mantendo a obrigatoriedade dos mesmos campos da adição.

Para apagar um cliente é necessário que este não tenha nenhum trabalho associado, caso contrário se se apagasse um cliente iria existir trabalhos sem clientes, o que não é possível.

Por tanto esse botão só está ativo caso não seja encontrado nenhum trabalho associado com o cliente. No entanto caso se apague todos os trabalhos associados com o cliente já se pode apagar o mesmo.

### Código da página

O código das páginas de gestão são das mais extensas, tanto por ter código para botões, como para as queries e validações.

As validações seguem sempre o mesmo padrão, que verificam todos os caracteres se são permitidos ou não, se não forem apagam ou substituem o caracter pelo permitido.

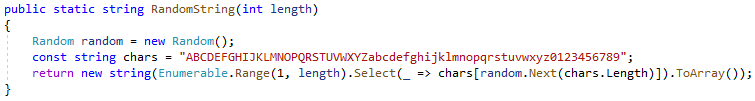
No início do programa, a primeira função que é executada é a de ligar á base de dados, que consiste simplesmente em carregar todos os dados necessários para a página, neste caso são os clientes.



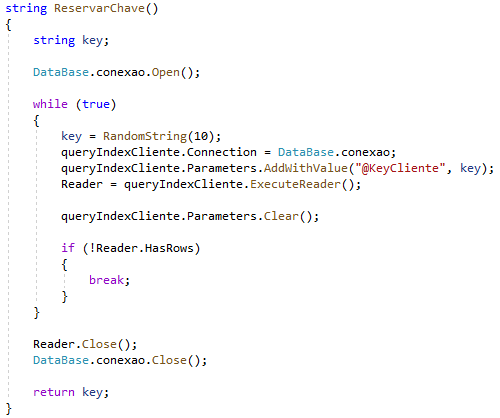
O código em cima permite ir á base de dados e carregar todos os campos do cliente, sendo que existe um campo chamado de “Contacto” que antes vai a uma função para ser mostrado o contacto preferível ou o contacto que é possível mostrar. Depois disso os dados são mostrados na listbox do lado esquerdo.

Nesta página assim como em todas as outras, existe uma função em que gera uma chave aleatória, esta chave é alfanumérica e gerada aleatoriamente, por várias razões.

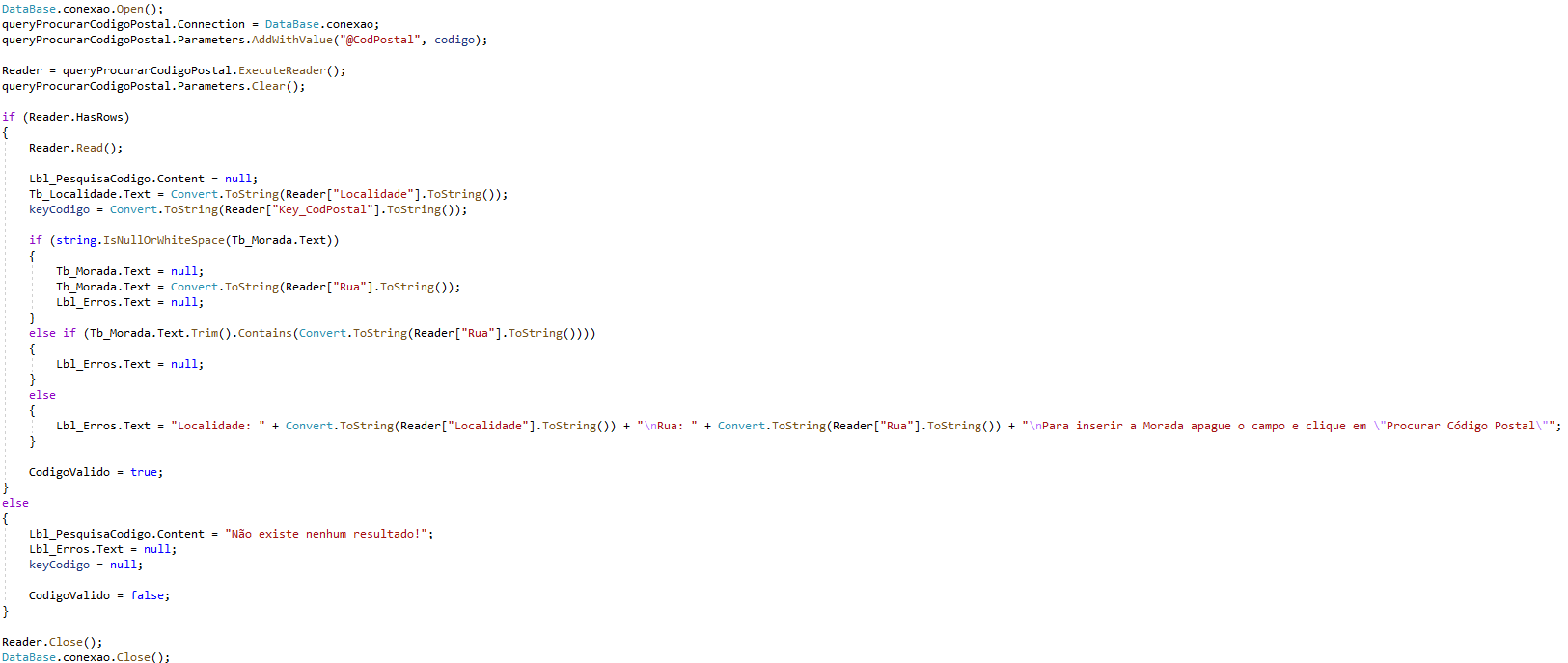
* Uma chave deste tipo com o tamanho de 10 caracteres permite um número muito maior de registos do que uma chave numérica que incrementa automaticamente.
* Uma chave numérica que incrementação automática pode chegar ao limite e não ser possível aumentar mais, embora este limite seja muito elevado, pode acontecer, e depois não seria possível utilizar mais o programa. Com a chave aleatória a capacidade de registos é maior e caso um registo seja apagado, essa chave está novamente livre.
* Permite chaves não sequenciais o que se fosse um serviço online era um novo patamar de proteção.



A função em cima é onde é gerada a chave com todos os caracteres que são possíveis usar, depois de gerada a chave a mesma é retornada a outra função.



Esta chave é onde se verifica que a chave ainda não existe na base de dados, na tabela de clientes, caso exista vai ser pedida uma nova chave até encontrar uma chave que seja possível usar.



O botão para procurar o código postal vai procurar entre 220000 registos um que tenha o mesmo código postal e vai atribuir a localidade e a rua aos dois campos. E a chave vai ser atribuída ao campo do código postal na tabela de clientes. Que é o que o código a cima faz.

Os registos de códigos postais encontrei em uma pergunta feita em um fórum português (portugal-a-programar.pt), em que havia um link com uma base de dados em Access de códigos postais disponibilizada pelos CTT em 2009.

Em relação para validar o nome de cliente se já existe na base de dados, foi feita uma simples query, para pesquisar na base de dados. Mas como este valor tambem tem de ser validado na edição foi preciso verificar se a chave do cliente era diferente do cliente que estávamos a editar, caso fosse diferente, então é um cliente com nome repetido e não é possível inserir na base de dados.



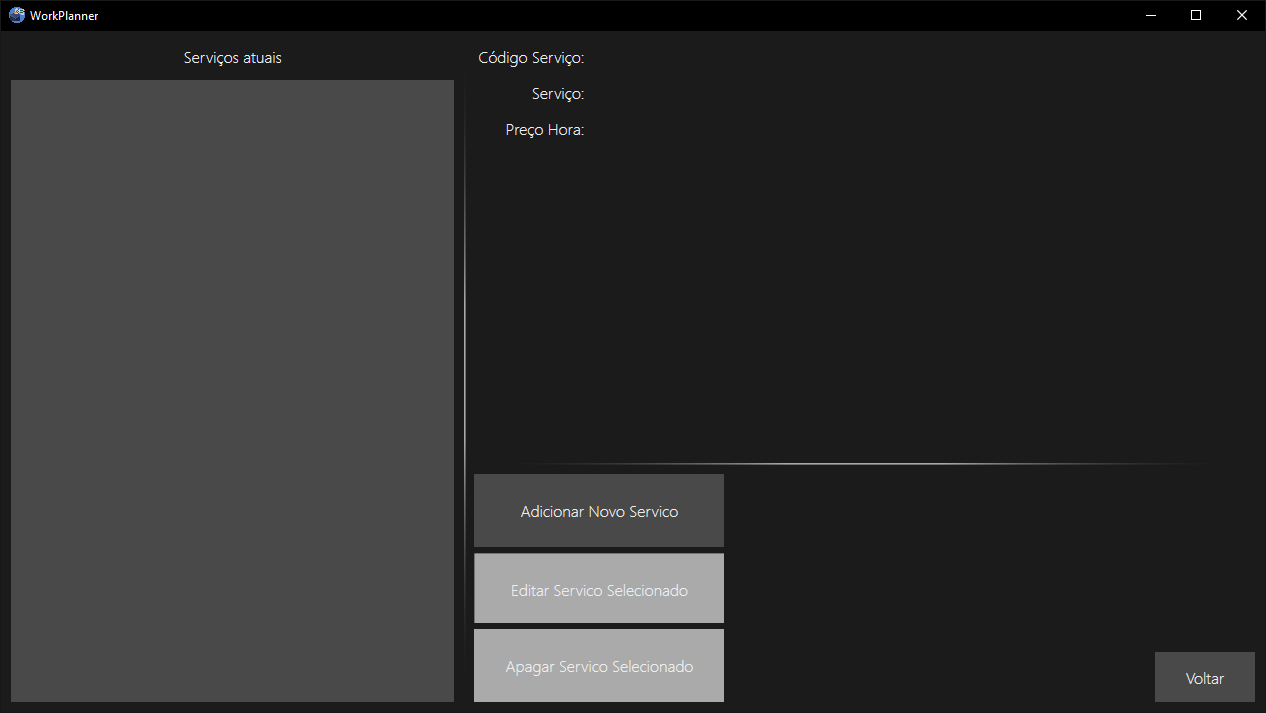
Depois da query feita é por este processo que passa, primeiro verifica se existem resultados, caso não existam é porque não existe nenhum nome igual na base de dados, caso contrário verifica-se se o código do cliente é igual, se for então não é uma duplicação, se não for igual então o nome está em duplicado.

As validações são processos simples onde simplesmente o programa verifica todos os caracteres se são válidos ou não, caso não sejam esses mesmos caracteres são apagados ou alterados.

O processo de adição ou edição do cliente é simples, apenas carregando a chave do código postal (caso exista), e inserindo na base de dados. Ambos os sistemas funcionam de forma semelhante, no entanto a edição atualiza os campos em vez de adicionar.

## Gerir Serviços

Assim como a página anterior, esta tem o design semelhante, com a lista dos serviços do lado esquerdo e os campos de edição do lado direito. Esta página é a que tem menos campos para editar, no entanto é importante pois para criar um trabalho é preciso ter serviços.

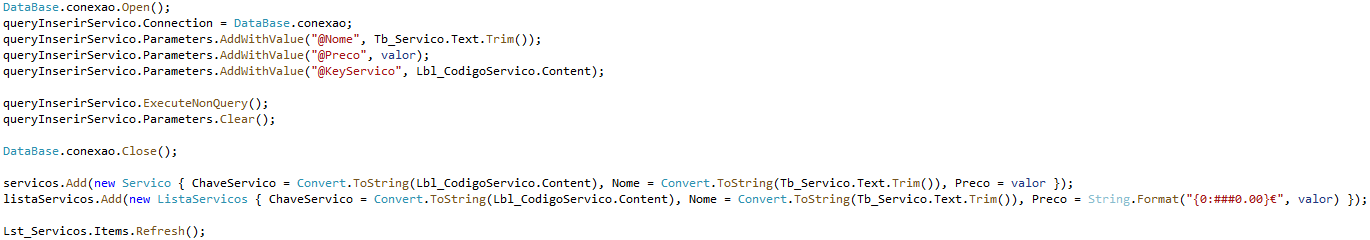


Assim como na página de clientes apenas dá para apagar o serviço caso não esteja associado a nenhuma tarefa.

Todos os campos têm validação, assim como no nome do serviço como no preço. O preço apenas permite números, uma virgula que não é no início e o símbolo do euro, ou não.

### Código da página

Esta página tem o mesmo funcionamento que a de clientes, onde carrega todos os serviços, adiciona na lista, caso se carregue em adicionar serviço, o programa vai gerar um código e verificar que ainda não existe. Após inserir o nome do serviço e de sair da textbox ou passado um segundo, a aplicação verifica se o serviço já está guardado ou não.



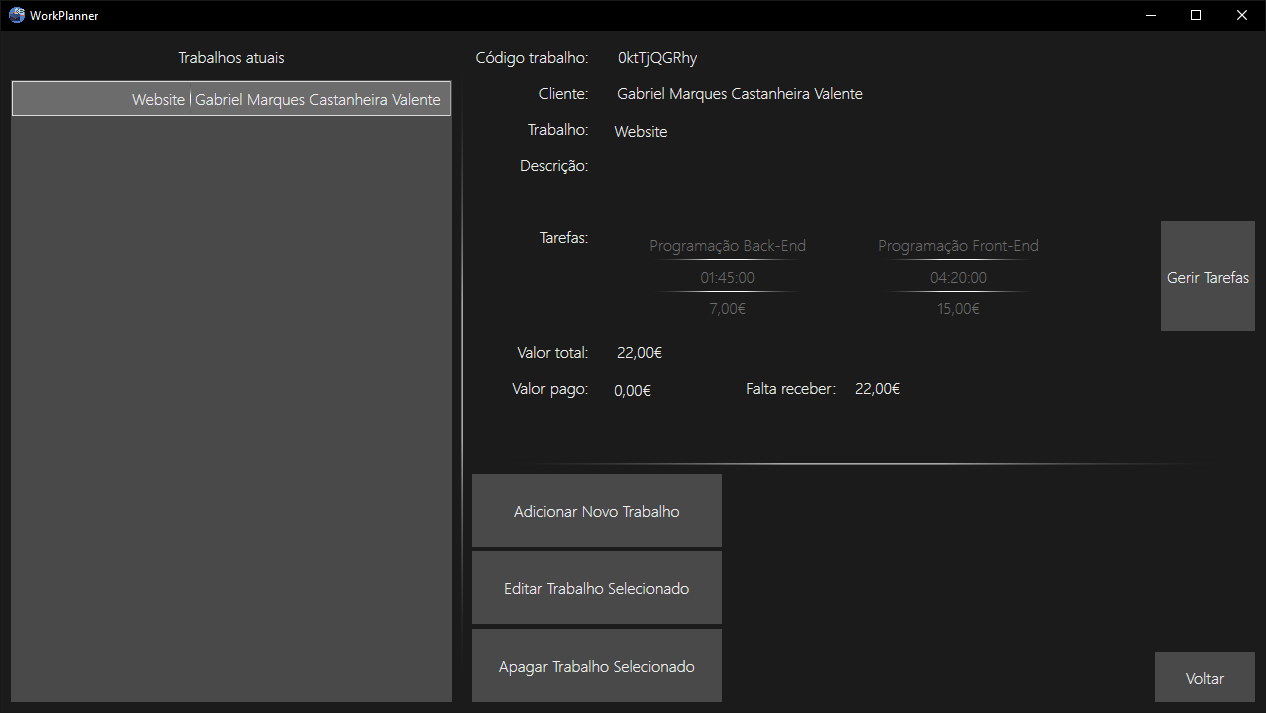
O código anterior é semelhante ao que está na página de clientes. Este código adiciona os valores á query e executa-a. Caso não dê problema, os dados são adicionados ás listas que são semelhantes, e a listagem é atualizada.

Esta imagem, assim como as anteriores, não mostram as formas de evitar erros, mas antes disso existe um if caso não tenha os dados, e um try catch para execução da query.

A função de edição é igual só que em vez de inserir dados novos atualiza na base de dados e na listagem.

## Gerir Trabalhos

Esta página pode ser das mais diferentes, das duas anterior, pois esta tabela tem informações de todas as outras trabelas. Tem informações de cliente, de tarefas e as tarefas tem de tempos.



Nesta página os campos editáveis são, Cliente, Trabalho, Descrição, Valor pago, os restantes campos ou são editados em outra página ou são calculados automaticamente.

O campo do cliente dá para escrever o nome, que aparece a lista com as várias opções, ou então dá para selecionar da lista. O trabalho e a descrição são campos de texto normais, sendo que o primeiro é o nome do trabalho, e a descrição é apenas texto corrido até 512 caracteres. Todos estes com as mínimas validações.

As tarefas são preenchidas em uma página seguinte, que ao adicionar ou selecionar um trabalho aparece um botão para abrir essa página.

O valor total é calculado a partir das tarefas que pertencem a este trabalho. E o valor pago é o que já recebemos do cliente.

### Código da página

Esta página tem algum código mais diferente do anterior, pois sempre que a página é carregada, vai atualizar todos os dados, pois se formos editar as tarefas e voltarmos á página é necessário apresentar as alterações. Para isso utilizei uma função que é “Loaded” que executa sempre que a página carrega.

Ao carregar existem duas opções, caso esteja algum item selecionado na lista, apenas serão atualizadas as tarefas do trabalho em questão.



O código em cima permite carregar as tarefas todas do trabalho selecionado, o que permite ter as alterações feitas na outra página.



O código seguinte faz os cálculos dos tempos de cada tarefas para adicionar no trabalho, e dessa forma mostrar o valor final do trabalho selecionado.

Este código é executado tanto quando se volta á página depois de editar as tarefas assim como quando selecionamos um trabalho a partir da lista.

Caso não esteja nenhum item selecionado é carregado um código parecido só que, no entanto, este carrega tambem a lista de trabalhos. E o mesmo que carregou anteriormente.

Se formos adicionar um novo trabalho acaba por gerar uma chave, tal como nas duas últimas páginas.

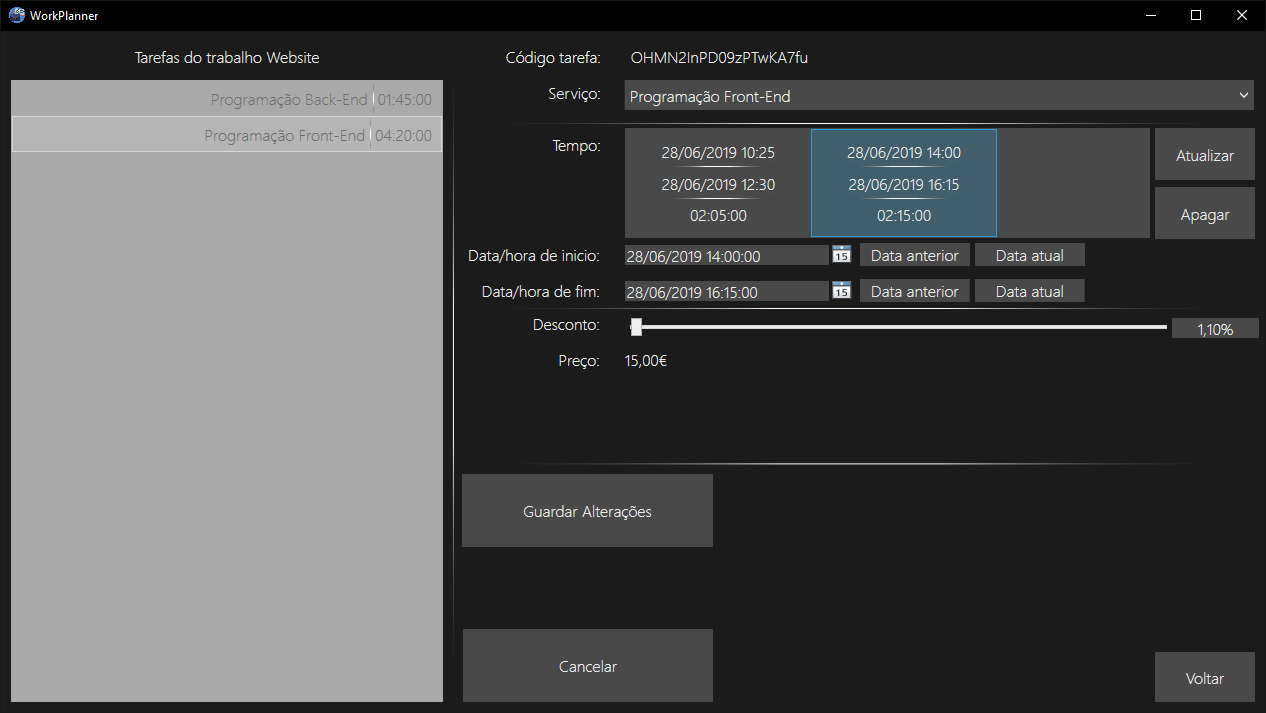
Se durante a adição do trabalho formos adicionar uma tarefa é preciso que todos os campos estejam válidos, pois quando se muda de página é necessário que o trabalho esteja guardado. Para associar as tarefas ao trabalho.

Tal é feito como uma das queries antes mostradas.

## Gerir Tarefas

Esta foi uma das páginas que deu mais problemas, por causa de validações que, no entanto, foi possível resolver após colocar várias questões no StackOverflow.

Esta página só dá para aceder a partir da página de trabalhos.



Esta página está a ser usada para um trabalho com o nome de “Website”, dá para ver pois em cima da lista aparece “Tarefas do Trabalho Website”.

Nesta página adiciona-se o tipo de serviço por uma lista que dá para pesquisar, e adiciona-se os tempos.

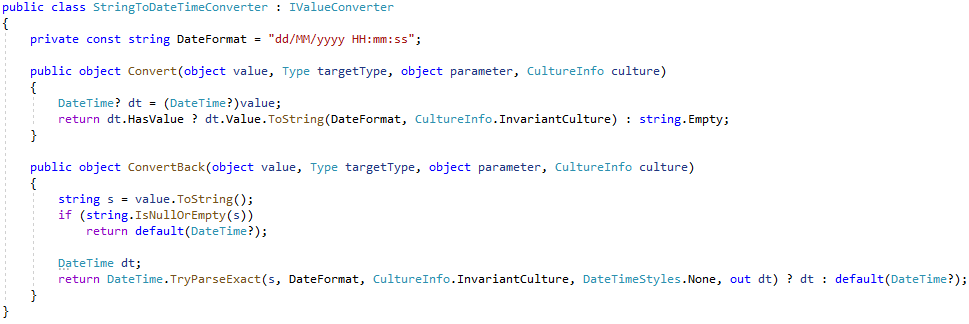
É possível adicionar mais que um intervalo de tempos por tarefa, por exemplo das 08:00 as 09:00 e das 11:00 as 12:00 do mesmo serviço, assim como dá para ter vários serviços no mesmo trabalho.

Ao adicionar ou editar dá para colocar a data/hora atual e posteriormente corrigir, como apagar os segundos, ou alterar os minutos, qualquer alteração é possível, tambem dá para inserir um intervalo de tempo sem data de fim, este intervalo vai ser considerado como zero, assim apenas depois de terminar essa tarefa insere-se a data de fim.

Todos estes valores são atualizados ao alterar qualquer um dos valores, tanto como serviço, como horas, ou até mesmo desconto.

### Código da página

Um dos problemas desta página foram as validações da data /hora, pois eu queria que fosse possível selecionar a data a partir do calendário e também queria que desse para escrever, para isso tive de ativar um dos elementos no calendário, e aplicar uma formatação. Com os erros que encontrei foi necessário recorrer ao StackOverflow para ter ajuda em relação a este problema.



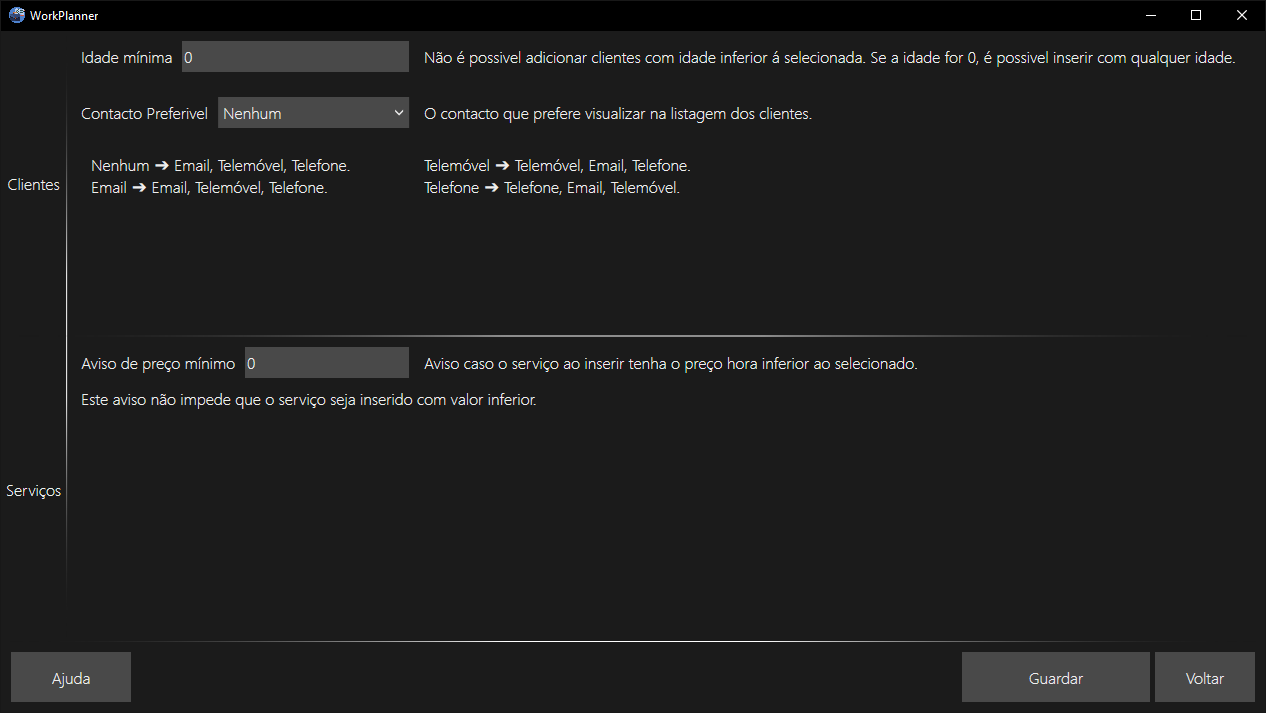
O código seguinte é o que permite a formatação do campo com data e hora e dar para escrever ou ter data nula. Pois um dos problemas era após ter uma data válida não era possível ter uma data nula.

O resto do funcionamento é semelhante ao das outras páginas. Somo o tempo da tarefa que então é multiplicado pelo valor hora e pelo desconto.

Após inserir os tempos na lista, ao guardar a tarefa estes tempos tambem serão guardados, editados ou apagados.

## Definições

Nesta página que foi referido no início do relatório, tem apenas três definições, que estava á espera de poder definições para alterar, mas devido a certos problemas com base de dados e o próprio funcionamento do programa não foi possível.



As três definições são a idade mínima que impede de inserir clientes com idade inferior á selecionada, caso a idade seja de 0, dá para inserir um cliente com qualquer idade.

A definição seguinte é o contacto preferível, que neste caso tem um pequeno esquema a mostrar como que são, esta definição é para mostrar o contacto na listagem dos clientes.

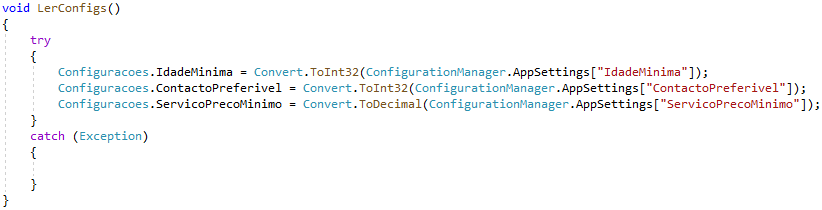
A última definição é um aviso que ocorre na página de serviços onde recebemos um aviso caso seja inserida um preço hora inferior ao selecionado aqui.

Existe três botões no fundo da página, o botão mais á esquerda permite ie para a página do GitHub onde o projeto foi criado, e direciona para a parte de problemas, onde o utilizador pode explicar o problema que lhe aconteceu para que seja corrigido numa próxima atualização.

Os dois botões á direita permitem guardar as alterações e voltar á página principal, ou simplesmente voltar á página principal sem guardar nenhuma alteração.

### Código da página

Esta página tem um código diferente ao das outras páginas, pois este não trabalha com base de dados, mas sim com o sistema de configurações do próprio programa.



O código em cima permite ler as configurações que estão no sistema, caso não existam configurações guardadas vai dar erro, e dessa forma não vai haver dados para apresentar, no entanto o valor “default” de uma variável tipo int ou decimal é 0, por isso serão esses valores a serem apresentados.



O código anterior vai abrir o ficheiro das configurações, sem nenhum tipo de configuração de utilizador. De seguida vai remover as definições guardadas, pois caso se escreva sem apagar as anteriores vão existir seis configurações guardadas em vez de três, sendo elas tambem repetidas.

Após estar tudo adicionado é guardado sem nenhum tipo de configuração especifica. O ficheiro é atualizado e são lidas novamente as definições para se poder utilizar no programa.

# Conclusão

Para concluir posso afirmar que, o desenvolvimento desta aplicação foi extremamente importante para aprender estas novas linguagens e melhorar os meus conhecimentos anteriores. Esta aplicação foi um grande desafio onde tive de desenvolver várias partes que não imaginei ter de trabalhar.

Posso afirmar tambem que depois do desenvolvimento deste programa posso voltar a abraçar outro projeto que utilize esta tao famosa linguagem de programação, pois muitas aplicações são desenvolvidas tambem em C# como C++ ou C, por tanto sei que tenho um excelente início, para o meu futuro profissional e/ou académico.

Os maiores desafios que encontrei com o desenvolvimento do projeto foram, alguns problemas com a base de dados, pois foi a primeira vez que utilizei uma base de dados local, sem utilizar outro software, como o XAMPP, e esta base de dados foi guardada como ficheiro, e por estar localizada em uma pasta errada sempre que iniciava o programa os dados era eliminados.

Outro dos problemas que tive foi a criação dos gráficos pois, a biblioteca que utilizei tinha alguns problemas, que devem ser resolvidos na próxima versão, então foi necessário encontrar uma forma de dar a volta.

Por último, a formatação dos campos de data e hora, na página de gerir tarefas, que causou alguns problemas, por não estar a conseguir inserir datas nulas.

De resto, os outros problemas que tive foram os ditos normais no desenvolvimento de uma aplicação que tenho de ser capaz de resolver sem grande demora.

# Webgrafia

LiveCharts - <https://github.com/Live-Charts/Live-Charts>

portugal-a-programar.pt - <https://www.portugal-a-programar.pt/forums/topic/64939-base-dados-todas-as-moradas-cp-de-portugal/>

Base de dados CTT - <https://app.box.com/s/di25740biozals6hnonj>

StackOverflow - <https://stackoverflow.com/questions/56621093/disable-datepicker-textbox-conversion-validation-on-focus-lost>