**Roteiro de Teste**

**Projeto: Equipamentos**

**Localização: Classe criptografia**

**Objeto de Teste: Gerar hash**

**Caso de Teste:**

Descrição:

Deve gerar um hash para uma string senha

Pré-condição:

Implementar o método gerarHash na cadastroUsuario

Procedimento: Simular um usuário fake.

Verificar através de uma expressão regular se a senha foi cadastrada como um hash no padrão Bcrypt.

Resultado esperado:  A senha que foi digitada originalmente sem criptografia deve estar criptografada no banco seguindo o padrão Bcrypt

**Localização: Classe Usuário**

**Objeto de Teste: Cadastrar Usuário**

**Caso de Teste:**

Descrição:  Deve ser adicionado as novas informações de um usuário no banco.

Pré-condição: Informações Fakes para cadastro

Procedimento: As informações vão ser preenchidas nos seus devidos campos e depois disso quando o método for iniciado as informações devem ser salvas no banco.

Resultado Esperado: Um novo usuário foi adicionado ao banco de dados.

**Localização: Controller Login**

**Objeto de Teste: Login Usuário**

**Caso de Teste:**

Descrição:  Deve ser verificado se ao adicionar as informações para login se é possível gerar um token.

Pré-condição: Email e Senha do banco.

Procedimento: Será preenchido email e senha na API e será executado o método de Login e será observado se ele gerara um token.

Resultado esperado: Foi gerado um token válido para o cliente.

**Localização: Classe Usuário**

**Objeto de Teste: Buscar usuários**

**Caso de Teste:**

Descrição: Deve ser possível buscar um usuário através de um ID válido.

Pré-condição: Id do usuário a ser buscado.

Procedimento: Pegar o Id digitado e preenchê-lo para realizar uma busca no banco por um usuário com aquele Id

Resultado esperado: O usuário que tem o mesmo id que foi digitado.

**Localização: Classe Usuário**

**Objeto de Teste: Preencher campo Id do método de buscar usuários**

**Caso de Teste:**

Descrição: Ao ser inserido um id o valor deve ser recebido no método de busca.

Pré-condição: Id do usuário a ser buscado.

Procedimento: Pegar o Id digitado e preenchê-lo e verificar se foi preenchido corretamente.

Resultado esperado: O campo Id do método foi preenchido.

**Localização: Classe Usuário**

**Objeto de Teste: Buscar usuários**

**Caso de Teste:**

Descrição: Deve ser possível buscar um usuário através de um ID válido.

Pré-condição: Id do usuário a ser burcado.

Procedimento: Pegar o Id digitado e preenchê-lo para realizar uma busca no banco por um usuário com aquele Id

Resultado esperado: O usuário que tem o mesmo id do digitado.

**Localização: Classe Vaga**

**Objeto de Teste: Atualizar Vaga**

**Caso de Teste:**

Descrição: Ao ser inserido um id e um novo valor pro status da vaga ele deve ser atualizado quando o banco rodar.

Pré-condição: Id da vaga a ser buscado e novo status.

Procedimento: Pegar o Id digitado e preenchê-lo buscando uma vaga no banco de dados e depois atualizar o status dela.

Resultado esperado: O status da vaga selecionada foi alterado com sucesso.

**Localização: Classe Vaga**

**Objeto de Teste: Listar Todas Vaga**

**Caso de Teste:**

Descrição: Ao ser inserido um id de um bicicletário será possível listar todas as vagas de um bicicletário especifico.

Pré-condição: Id do bicicletário.

Procedimento: Pegar o Id digitado e preenchê-lo buscando vagas com o campo IdBicicletario semelhantes.

Resultado esperado: Uma lista de vagas com o IdBicicletario similar ao recebido.

**Localização: Classe Vaga**

**Objeto de Teste: Atualizar Vaga**

**Caso de Teste:**

Descrição: Ao ser inserido um id e um novo valor pro status da vaga ele deve ser atualizado quando o banco rodar.

Pré-condição: Id da vaga a ser buscado e novo status.

Procedimento: Pegar o Id digitado e preenchê-lo buscando uma vaga no banco de dados e depois atualizar o status dela.

Resultado esperado: O status da vaga selecionada foi alterado com sucesso.

**Localização: Classe Reserva**

**Objeto de Teste: Cadastrar Reserva**

**Caso de Teste:**

Descrição: Ao ser recebido um novo objeto do tipo reserva preenchido ele deve poder ser cadastrado no banco.

Pré-condição: Informações fake para o cadastro.

Procedimento: Pegar o novo objeto preenchido e realizar o cadastro dele no banco de dados.

Resultado esperado: A criação de uma nova reserva.

**Localização: Classe Reserva**

**Objeto de Teste: Atualizar Reserva**

**Caso de Teste:**

Descrição: Ao ser recebido novas informações da reserva é possível atualizá-la.

Pré-condição: Informações Horário de destravamento da trava, Preço e Status do Pagamento.

Procedimento: Atualizar a reserva a partir dos dados recebidos.

Resultado esperado: A Atualização de uma reserva existente.

**Localização: Classe Reserva**

**Objeto de Teste: Listar Minhas Reserva**

**Caso de Teste:**

Descrição: Será possível listar somente as reservas do usuário que estiver logado.

Pré-condição: O usuário deve estar logado na aplicação.

Procedimento: Listar as reservas do usuário que estiver logado no momento.

Resultado esperado: Uma lista de reservas.

**Localização: Classe Reserva**

**Objeto de Teste: Preencher o capo do Id do usuário automaticamente a partir do Token**

**Caso de Teste:**

Descrição: Será preenchido o valor IdUsuario no campo listar minhas a partir do token.

Pré-condição: O usuário deve estar logado na aplicação.

Procedimento: Colocar o valor do campo como o valor no token.

Resultado esperado: O campo IdUsuario foi preenchido apartir do token corretamente.

**Localização: Classe Bicicletário**

**Objeto de Teste: Cadastrar Bicicletário**

**Caso de Teste:**

Descrição: Ao ser recebido um novo objeto do tipo um novo objeto Bicicletário preenchido ele deve poder ser cadastrado no banco.

Pré-condição: Informações fake para o cadastro.

Procedimento: Pegar o novo objeto preenchido e realizar o cadastro dele no banco de dados.

Resultado esperado: A criação de um novo bicicletário.

**Localização: Classe Bicicletário**

**Objeto de Teste: Listar Bicicletário**

**Caso de Teste:**

Descrição: Deve ser possível listar todos os bicicletário cadastrados.

Pré-condição:

Procedimento: Executar o método para que seja feita uma lista com todos os bicicletários cadastrados.

Resultado esperado: Lista de bicicletário cadastrados.

**Localização: Classe Bicicletário**

**Objeto de Teste: Buscar Bicicletário**

**Caso de Teste:**

Descrição: Ao receber um id, deve ser realizado o método para buscar reservas com o mesmo id que foi digitado.

Pré-condição: Id do bicicletário a ser buscado.

Procedimento: Pegar o ID digitado e realizar uma busca para ver qual bicicletário tem o id similar aquele.

Resultado esperado: Um bicicletário com id similar ao digitado.