

# Guia de Instalação do programa desenvolvido

---

## 1. Criar as tabelas referentes à Simulação de Crédito, ao Estado de Crédito e à Periodicidade

### 1.1. Abrir o MySQL Workbench

### 1.2. Conectar ao MySQL Server (Local Instance)

### 1.3. Copiar e correr o seguinte código:

```
USE youcredit;

DROP TABLE IF EXISTS cr_simulacao_credito;
DROP TABLE IF EXISTS cr_estado_credito_simulacao;
DROP TABLE IF EXISTS cr_periodicidade_estado_prestacao;

CREATE TABLE cr_estado_credito_simulacao (
    estado_id INT AUTO_INCREMENT,
    estado_simulacao VARCHAR (15) NOT NULL,
    descricao VARCHAR (15) NOT NULL,
    data_log TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    utilizador_log INT NOT NULL,
    estado_log VARCHAR(1) NOT NULL DEFAULT "A",

    PRIMARY KEY (estado_id),
    CONSTRAINT cr_estado_credito_simulacao_chk_estado_log CHECK (estado_log="A"
OR estado_log="I")
);

CREATE TABLE cr_periodicidade_estado_prestacao (
    periodicidade_id INT AUTO_INCREMENT,
    periodicidade VARCHAR(15) NOT NULL,
    descricao VARCHAR(15) NOT NULL,
    data_log TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    utilizador_log INT NOT NULL,
    estado_log VARCHAR(1) NOT NULL DEFAULT "A",

    PRIMARY KEY (periodicidade_id),
    CONSTRAINT cr_periodicidade_estado_prestacao_chk_estado_log CHECK
(estado_log="A" OR estado_log="I")
);

CREATE TABLE cr_simulacao_credito (
    simulacao_id INT AUTO_INCREMENT,
    referencia VARCHAR(50),
    flag_credito BOOLEAN NOT NULL,
    data_solicitacao DATE,
    data_decisao DATE,
    data_inicio DATE NOT NULL,
    data_fim DATE,
    data_avaliacao DATE,
    data_alteracao_estado_credito DATE,
    total_solicitado DOUBLE NOT NULL,
```

```

total_concedido DOUBLE,
total_possivel DOUBLE NOT NULL,
total_capital DOUBLE,
total_juro DOUBLE NOT NULL,
total_despesa DOUBLE NOT NULL,
total_imposto DOUBLE NOT NULL,
descricao_objeto VARCHAR(50),
duracao INT,
scoring INT NOT NULL,
data_alteracao_estado_simulacao DATE NOT NULL,
parecer BOOLEAN NOT NULL,
estado_id INT,
periodicidade_id INT,
produto_id INT,
entidade_id INT,
entidadeavalista VARCHAR(50),
existe_crm VARCHAR(1),
data_log TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
utilizador_log INT NOT NULL,
estado_log VARCHAR(1) NOT NULL DEFAULT "A",

PRIMARY KEY (simulacao_id),

FOREIGN KEY (estado_id)
    REFERENCES cr_estado_credito_simulacao (estado_id),

FOREIGN KEY (periodicidade_id)
    REFERENCES cr_periodicidade_estado_prestacao (periodicidade_id),

FOREIGN KEY (produto_id)
    REFERENCES pr_produto (produto_id),

FOREIGN KEY (entidade_id)
    REFERENCES en_entidade (entidade_id),

CONSTRAINT cr_simulacao_credito_chk_estado_log CHECK (estado_log="A" OR
estado_log="I")
);

```

## 2. Criar os Estados necessários ao bom funcionamento do programa

Edit > Preferences > SQL Editor > Other > Safe Updates (rejects UPDATES and DELETES with no restrictions)

```

use youcredit;

DELETE FROM cr_estado_credito_simulacao;
INSERT INTO cr_estado_credito_simulacao
(estado_id,estado_simulacao,descricao,data_log,utilizador_log,estado_log)
VALUES (1,"Registado","Registado",current_timestamp(),0,"A");
INSERT INTO cr_estado_credito_simulacao
(estado_id,estado_simulacao,descricao,data_log,utilizador_log,estado_log)
VALUES (2,"Para Aprovação","Para Aprovação",current_timestamp(),0,"A");
INSERT INTO cr_estado_credito_simulacao
(estado_id,estado_simulacao,descricao,data_log,utilizador_log,estado_log)

```

```
VALUES (3,"Sem Efeito","Sem Efeito",current_timestamp(),0,"A");
INSERT INTO cr_estado_credito_simulacao
(estado_id,estado_simulacao,descricao,data_log,utilizador_log,estado_log)
VALUES (4,"Aprovado","Aprovado",current_timestamp(),0,"A");
INSERT INTO cr_estado_credito_simulacao
(estado_id,estado_simulacao,descricao,data_log,utilizador_log,estado_log)
VALUES (5,"Não Aprovado","Não Aprovado",current_timestamp(),0,"A");
INSERT INTO cr_estado_credito_simulacao
(estado_id,estado_simulacao,descricao,data_log,utilizador_log,estado_log)
VALUES (6,"Sem Interesse","Sem Interesse",current_timestamp(),0,"A");

select * from cr_estado_credito_simulacao;
```

## 4. Criar Stored Procedures

```
#use youcredit;
#DROP PROCEDURE IF EXISTS AtualizarEstadoSimulacao

DELIMITER $$
CREATE PROCEDURE AtualizarEstadoSimulacao(IN ytempo INTEGER, IN estado INTEGER,
estado_nome VARCHAR(20))
BEGIN
DECLARE xfim_cursor INTEGER DEFAULT 0;
DECLARE xestado VARCHAR(1);
DECLARE xestado_id INTEGER;
DECLARE xqtd_registos INTEGER;

DECLARE xCursorSimulacoes CURSOR FOR
SELECT estado_id
FROM cr_simulacao_credito
WHERE datediff(curdate(),data_alteracao_estado_credito) > ytempo; # 5 ou 10
dias
DECLARE CONTINUE HANDLER FOR NOT FOUND SET xfim_cursor = 1;

OPEN xCursorSimulacoes;
SET xqtd_registos = 0;
ObterSimulacao: LOOP
FETCH xCursorSimulacoes INTO xestado_id;

IF xfim_cursor = 1 THEN
LEAVE ObterSimulacao;
END IF;
UPDATE cr_simulacao_credito
SET estado_id = estado, classeestadocredito_estado_simulacao = estado_nome
WHERE estado_id = xestado_id;
SET xqtd_registos = xqtd_registos + 1;
END LOOP ObterSimulacao;
CLOSE xCursorSimulacoes;
END$$
DELIMITER ;
```

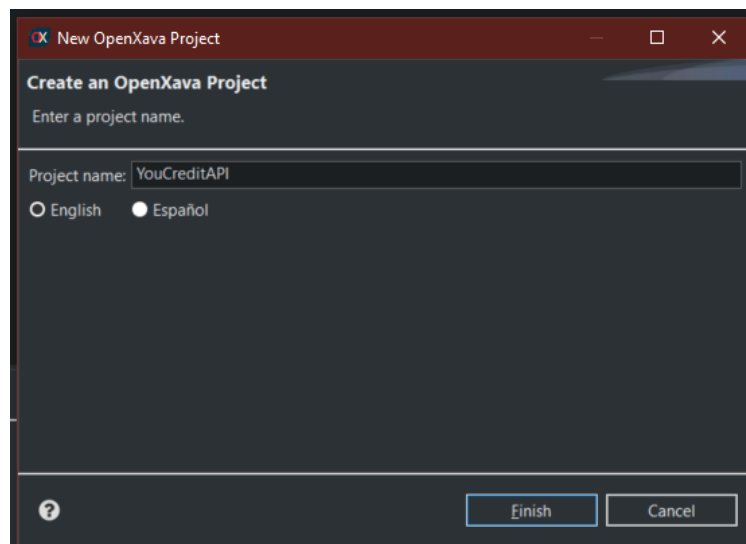
## 5. Iniciar Eventos

```
DROP EVENT IF EXISTS `Evento_AtualizarEstado_5dias`;
CREATE EVENT `Evento_AtualizarEstado_5dias`
ON SCHEDULE EVERY 60 minute
STARTS current_timestamp()
DO CALL AtualizarEstadoSimulacao(5,3,"Sem Efeito");

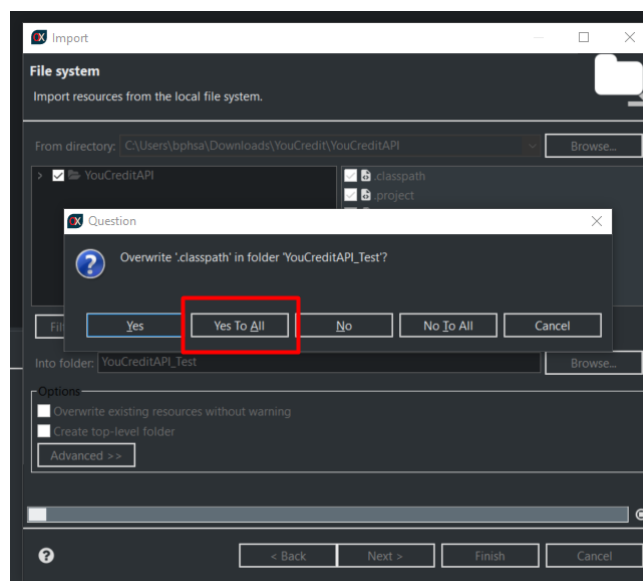
DROP EVENT IF EXISTS `Evento_AtualizarEstado_10dias`;
CREATE EVENT `Evento_AtualizarEstado_10dias`
ON SCHEDULE EVERY 60 minute
STARTS current_timestamp()
DO CALL AtualizarEstadoSimulacao(10,6,"Sem Interesse");
```

## 6. Criar um novo OpenXava Project

Abaixo encontram-se os passos necessários à importação do YouCreditAPI. Terá de fazer o mesmo para o YouCreditFE e YouCreditBE.



### 6.1. Importar ficheiros recebidos



## 6.2. Mudar Password da BD



## 6.3. Alterar o PATH das variáveis Java-JSON e MYSQL JDBC

Remova as variáveis Java-JSON e MYSQL JDBC existentes no Java Build Path e volte a adicionar ambas através da opção "Add Variable".

